VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Data mining a SQL Server 2008

Data mining and SQL Server 2008

2010 Jakub Poulíček

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava*.

V Ostravě 16. dubna 2009 .....

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 16. dubna 2009 .....

Děkuji své vedoucí, Mgr. Pavle Dráždilové, za odborné rady a pomoc při vedení mé bakalářské práce a všem lidem, kteří mi pomohli.

Abstrakt

Cílem této bakalářské práce je obecné seznámení s principy, funkcemi a využitím dolování dat a jeho realizaci v SQL Serveru 2008. První část se zabývá popisem integračních služeb a metodikou uchovávání vícerozměrných dat. Druhá část popisuje teorii a implementaci jednotlivých používaných metod. Třetí část se pak zaměřuje na použití těchto metod a porovnání výhodnosti jejich aplikace na zadaná data.

Klíčová slova: Analýza dat, Dolování dat, Business Intelligence, SQL Server 2008

Abstract

The goal of this bachelor project is to show the principles, functions and usage of data mining and its realisation in SQL Server 2008. The first part is concerned in description of integration services and way of saving multidimensional data. The second part describes theory and implementation of used functions. The third part is focused on using these methods in SQL Server 2008 Developer Edition, comparing its efficiency of its aplications on obtained data.

Keywords: Data analysis, Data mining, Business Intelligence, SQL Server 2008

Seznam použitých zkratek a symbolů

BI	–	Bussiness inteligence
CSV	–	Comma Separated Values
DM	–	Data Mining
DSS	–	Decision-Support Systems
EM	–	Estimation Maximization
EIS	–	Eecutive information system
KDD	–	Knowledge Discovery in Databases
MIS	–	Management Information System
MS	–	Microsoft
OLAP	–	On-line Analytical Processing
OLTP	–	On-line Transaction Processing
XMLA	–	XML for Analysis

Obsah

Seznam obrázků

1 Úvod

Dolování dat je významnou, ne-li nejvýznamnější, složkou technologií business intelligence. Tyto technologie slouží pro převod surových historických dat firem na informace, případně znalosti, které jsou snadno interpretovatelné. Na základě takto získaných informací je pak možné optimalizovat firemní procesy, marketingové kampaně, atd.

Cílem mé práce bylo seznámit se s aplikací integračních a analytických služeb (tedy nástroji pro dolování dat) technologie MS SQL Servr 2008 a provést experimenty s aplikací metod dolování dat nad zadanou databází.

V první části své práce se věnuji obecnému popisu business intelligence, integračním službám a jejich využití a způsobu ukládání vicerozměrných dat.

Následující část je zaměřena na popis analytických služeb, které dává SQL Server 2008 k dispozici. V této části se u popisu každé metody nejprve věnuji popsání principů jejího způsobu analýzy a následuje popis toho, jak je konkrétně metoda implementována v SQL Serveru 2008. Na závěr popisu pak uvádím seznam parametrů metody, které je možno použít pro její optimalizaci.

Poslední kapitola obsahuje popis experimentů s integračními službami a samotným dolováním dat. Experimenty byly provedeny nad souborem obsahujícím přihlašovací data (logy) do systému Moodle, který slouží jako podpora pro e-learning. Pomocí integračních služeb tedy bylo potřeba data ze souboru korektně vybrat a uložit je do databáze serveru. Následovala aplikace jednotlivých analytických služeb na získanou databázi. Popisy experimentů s metodami sestávají z uvedení vstupního nastavení a konfigurace algoritmu testované metody, délky výpočtu, interpretace výsledku a vyhodnocení efektivity (resp. informačního přínosu) metody.

V závěru shrnuji získané poznatky a zkušenosti s nasazením integračních a aplikačních služeb a své dojmy s prací se systémem MS SQL Server 2008.

2 Business Intelligence v SQL Serveru 2008

Luboslav Lacko [3] definuje pojem Business Intelligence jako proces transformace dat na informace a následný převod těchto informací na poznatky. Petr Berka [4] interpretuje význam tohoto pojmu rovnicí:

$$business\ intelligence = artificial\ intelligence + business.$$

Luminita Hurbean [15] popisuje BI podle jeho funkce, jako nástroj umožňující organizacím extrakci užitečných informací z rychle rostoucího seznamu heterogenních zdrojů dat, včetně různých databázových platforem, datových skladů, datových trhů a e-business sytémů.

SQL Server 2008 disponuje v oblasti Business Intelligence nástroji pro integraci datových zdrojů, analýzu dat a report výsledné analýzy.

2.1 Integrace dat v SQL serveru 2008

Integrační služby MS SQL Serveru 2008 poskytují nástroj pro sjednocení heterogenních zdrojů dat. Může jít o klasické systémy relačních databází různých společností (Oracle, IBM, Microsoft,...), XML databáze, textové soubory s hrubými daty, podnikové systémy (např. SAP) nebo soubory tabulkových procesorů.

Integrační služby je možno navrhovat jako jednorázový nebo periodicky se vykonávající proces (viz [3]). Druhá možnost se týká především každodenně se aktualizujících datových skladů. Ve finále jsou data nahrána do zadaného druhu databáze.

2.1.1 Transakční databáze vs. Analytické databáze (OLAP)

Běžně používané transakční databáze jsou určeny (a optimalizovány) pro provádění nejrůznějších obchodních transakcí. Tomuto účelu je také uzpůsobena komplexnost a struktura údajů, které tyto databáze obsahují a na tomto poli dosahují vysokých výkonů. Pro provádění náročných analýz je však transakční model nevhodný. Pro komplexní analýzy jsou podstatně vhodnější multidimenzionální modely.

Vývoj přechodu od transakčně orientovaných systémů po analyticky orientované systémy měl několik fází [14].

- MIS - Vstupem těchto systémů jsou data transakčních systémů. MIS poskytovaly manažerům pravidelné strukturované zprávy, nicméně nebyly schopné asistovat manažerům v procesech rozhodování.
- DSS - Systémy DSS (systémy pro podporu rozhodování), již umožňují nasazení do procesu strategického rozhodování poskytováním výsledků komplexních analýz).
- EIS - Tyto systémy již umožňily manažerům přístup k dotazování databází - uživatelské rozhraní "zakrylo"syntaxi SQL. Nevýhodou však bylo omezení sady analytických metod na předem připravené šablony. Složitější konstrukce proto bylo nutné opět vytvářet převodem dotazu do jazyka SQL. EIS jsou již označovány termínem Business Intelligence.

- OLAP - Současné systémy OLAP se oproti EIS vyznačují intuitivním ovládáním a uživatelsky přívětivějším rozhraním - mnohdy dávají k dispozici nástroje pro vizualizaci výsledků analýz. Definice těchto systémů je volná, jde spíše o systémy založené na určitých principech. Poměrně dobře je charakterizuje 12 pravidel OLAP od Dr. Edgara Franka Codda (viz [3] - pro poněkud odlišná kritéria sytémů OLAP viz [4])

    1. *Multidimenzionální konceptuální pohled* - OLAP musí nabízet multidimenzionální model odpovídající potřebám.
    2. *Transparentnost* - Architektura výpočtů, podřízená databáze a technologie OLAP by měla pro být uživatele přehledná a umožňovat mu snadné použití front-end nástrojů.
    3. *Dostupnost* - Systém by měl přistupovat jen k datům, které jsou nutné pro provedení analýzy.
    4. *Konzistentní vykazování* - Růst databáze by neměl znatelně ovlivňovat rychlost analýzy.
    5. *Architektura klient-server* - Systém OLAP musí pracovat na základě architektury klient-server.
    6. *Generická dimenzionalita* - Dimenze musí být co do struktury a operačních schopností ekvivalentní.
    7. *Dynamické ošetření řídkých matic* - OLAP by měl přizpůsobovat své fyzické uspořádání konkrétnímu analytickému modelu, včetně optimalizace ošetření řídkých matic (viz Datové modely v OLAP).
    8. *Podpora pro více uživatelů* - Systém musí podporovat práci více uživatelů na jednom modelu.
    9. *Neomezené křížové dimenzionální operace* - Systém musí být schopen rozpoznat hierarchie dimenzí a vykonávat asociované kumulované kalkulace jak nad jednou, tak nad více dimenzemi současně.
    10. *Intuitivní manipulace s daty* - Uživatelské rozhraní by mělo umožňovat intuitivní manipulaci se systémem a analytickými službami.
    11. *Flexibilní vykazování* - Systém musí umožňovat analýzu pomocí intuitivní vizuální prezentace, na základě uspořádání řádků a sloupců.
    12. *Neomezené dimenze úrovně agregace* - OLAP by neměl zavádět umělé omezení počtu podporovaných dimenzí modelu.

2.1.2 Datové modely v OLAP

Pro systémy OLAP je typické, že na data pohlížejí jako na tzv. *datovou krychli* (data cube). Je běžné, že tyto "kostky"mají více než 3 rozměry (nejde tedy v pravém slova smyslu o krychle, ale spíše *hyperkrychle* (hypercubes)). Systémy využívající n-rozměrné krychle jsou označovány jako MOLAP (multidimensional OLAP).

Atributy jsou zde reprezentovány dimenzemi, záznamy pak tovří jednotlivé buňky krychle. Kromě samotných záznamů z operační databáze obsahují datové krychle také dílčí agregace, které především umožňují velmi rychlou odezvu na ad-hoc dotazy (viz [4]).

Záznamy v těchto krychlích se pak nacházejí na příslušných průsečících jednotlivých dimenzí. Tento způsob však vede k velice řídkému ukládání dat. Fyzická implementace se proto od logické implementace liší. Nejrozšířenější jsou tyto dva přístupy (viz [4]) - oba dva jsou k dispozici v OLE DB:

- *Hyperkrychle* - Tento přístup implementuje datový model tak, že všechny dimenze náleží jediné n-rozměrné krychli. Výhodou tohoto přístupu je jednoduchá a srozumitlená struktura.
- *Multikrychle* - V tomto modelu jsou data rozděleny v několika menších krychlích, z nichž každá má přiřazeno jen několik vlastních dimenzí. Tento přístup je sice složitější než hyperkrychle, to je však vyváženo úspornějším způsobem uložení dat.

Nevýhodou systémů MOLAP je jejich náročnost na datový server (data jsou uložena jednak standardním způsobem v relační databázi, jednak v multidimenzionální databázi). Tyto systémy proto nejsou vhodné pro dynamické aplikace a své využití nacházejí spíše ve středně velkých (5-10 milionů záznamů v dimenzi) statických aplikacích (např. analýza prodeje určitého produktu). Výchozí nastavení úložného módu (storage mode) v SQL Serveru 2008 je nastaveno právě na MOLAP (viz [7]).

Pro opravdu rozsáhlé databáze (kde má dimenze více než 10 milionů záznamů) jsou vhodnější systémy *ROLAP* (relational OLAP). Systémy ROLAP používají pro analýzu data z relačních datových skladů, uživateli je po zpracování těchto dat zpřístupněn multidimenzionální pohled - nedochází tak k vytváření redundanních dat jako u MOLAP. Data i metadata se ukládají do relační databáze a OLAP server z nich dynamicky generuje SQL příkazy pro získání uživatelem požadovaných dat.

2.1.3 Ukládání analytických dat v BI

V analytických systémech jsou data běžně ukládána následujícími dvěma způsoby.

Datový sklad (Data store) - Místo, kde jsou data určená pro analýzu uložena se označuje jako datový sklad. Nejznámější definice datového skladu pochází od tvůrce tohoto konceptu - W. H. Inmona (převzato z [3]): Datový sklad je podnikově strukturovaný depozitář subjektivně orientovaných, integrovaných, časově proměnných, historických dat použitých na získávání informací a podporu rozhodování. V datovém skladu jsou uložena atomická a sumární data.

Inmon definoval následující vlastnosti typické pro datové sklady:

- *subjektová orientovanost* - V datovém skladu jsou oproti produkčním databázím uchovávána pouze data použitelná pro strategické rozhodování.
- *integrace* - Označuje fakt, že názvy ukazatelů, měřítka a kódování jsou sjednocené.

- *časová proměnnost* - Fixace dat z produkční databáze - datový sklad je pravidelně aktualizován (off-line).

- *stálost* - Data uložená v datovém skladu nejsou analytickými dotazy nijak měněna.

Datový sklad zpravidla není vytvářen pro konkrétní analýzu.

Datové tržiště (Data mart) Datová tržiště jsou přesně specifikované podmnožiny datového skladu určené pro menší organizační složky firmy. Datové trhy mohou vznikat "zdola", kdy jsou jednotlivým firemním oddělením vytvořeny datové trhy a na závěr je vytvožen zašťiťující datový sklad, nebo "shora", kdy je nejprve vytvořen centrální integrovaný datový sklad a teprve z něj se odštěpí několik datových trhů.

Data zavedená v databázi, databázovém skladu, případně kostce jsou již vhodná pro aplikaci analytických služeb.

2.2 Proces hledání znalostí v databázích

Hledání znalostí v databázích (Knowledge Discovering in Databases) je proces, který se poprvé objevil na začátku 90. let - tehdy také vzrostla potřeba zpracovávat firemní data za účelem podpory podnikové strategie [4]. KDD je integrací statistických metod a metod umělé inteligence, která umožňuje realizovat analýzy nad rozsáhlými databázemi vedoucí k informacím, které jsou relevantní pro podporu strategického rozhodování.

Ussama Fayyad [6] popisuje proces hledání znalostí v databázích, jako vývoj metod a technik, které mají dát datům význam. Problémem, který má KDD řešit pak je převod dat nízké úrovně (typicky rozsáhlé, lidskými silami obtížně zpracovatelné databáze) do jiné formy, která může vykazovat větší míru kompaktnosti, abstrakce, případně použitelnosti.

Pojem *dolování dat* (data mining) je pak označením aplikace konkrétních algoritmů extrahujících z databáze datové vzory - jde tedy pouze o část KDD procesu (viz níže). Data mining (resp. KDD) má široké využití, ať už jde o vědecký výzkum, průmysl, finančnictví, zdravotnictví nebo marketingové analýzy.

Proces dobývání znalostí z databází dále Fayyad rozděluje do následujících fází:

1. *Porozumění aplikační doméně.*

2. *Vytvoření cílové množiny dat* - tzn. výběr množiny proměnných na které se výzkum bude provádět.

3. *Pročišťování a předzpracování dat* - odstranění šumu, výběr strategií pro nakládání s chybějícími datovými poli, posouzení vhodnosti použití časových řad.

4. *Redukce dat a projekce* - nalezení znaků použitelných pro reprezentaci dat v závislosti na cílech úlohy.

5. *Spojení cíle KDD s vhodnou DM metodou* - sumarizace, klasifikace, regrese, atd.

6. *Analýza a výběr hypotézy* - výběr konkrétního algoritmu a metod selekce.

7. *Samotný DM proces* - vyhledávání vzorů v příslušné množině dat, převod výsledků do příslušné formy (shluky, stromy, atd.).

8. *Interpretace vydolovaných vzorů* - je možné, že v rámci iterace procesu dojde k návratu do kteréhokoliv předchozího bodu, spadá sem také vizualizace výstupu.

9. *Nakládání s nabytou znalostí* - přímé použití znalosti, začlenění do jiného systému, kontrola správnosti.

Podrobnému popisu jednotlivých metod a algoritmů dolování dat implementovaných v SQL Serveru 2008 se věnuji ve zvláštní kapitole.

2.3 Reportovací služby

Výsledky procesu dolování dat je v SQL Serveru 2008 možno prezentovat pomocí tzv. *reportovacích služeb*. Tyto služby slouží zejména pro zpřístupnění výsledků analýz, resp. podmnožin těchto výsledků, skupinám analytiků, případně přímo "konzumentů informací"(viz. [3]).

Ve své práci se reportovacími službami nezabývám - pro pouhé testování analytických služeb nemají využití.

3 Algoritmy pro dolování dat v SQL Serveru 2008

MS SQL Server 2008 poskytuje pro procesy dolování dat mnoho nástrojů. Výběr a aplikace konkrétního nástroje závisí na typu a parametrech řešeného problému.

Implementované metody (resp. konkrétní algoritmy) můžeme rozdělit podle základního způsobu, jakým pracují na několik skupin. Jde o asociační metody, časové řady, metody shlukovací, rozhodovací stromy, metodu Bayesovu a neuronové sítě[1].

3.1 Popis DM metod v SQL Serveru 2008

Stati pojednávající o jednotlivých metodách, které SQL Server 2008 používá jsou rozděleny na dvě části. V první části je popsána stručná historie zdrojů dané metody, obecně platné principy a teoretické zázemí.

Druhá část má poskytnout detailní vhled do principů funkce metody, tak, jak je implementována v SQL Serveru 2008, tj. popis konkrétních algoritmů, parametrů funkcí, které jsou k dispozici a bližších specifikací pro užití dané metody.

3.2 Možnosti nastavení DM modelu

SQL Server 2008 umožňuje nastavovat tzv. *modelové značky* (Modeling Flags), pomocí kterých je možné dodatečně upřesnit parametry DM modelu a lépe tak přizpůsobit jeho charakteristiky danému problému (např. omezením množiny dat, se kterou DM algoritmus bude zacházet, atd.).

Jedná se o tyto značky:

- *NOT NULL*

  - Určuje že záznamy takto označeného sloupce nesmí obsahovat prázdnou hodnotu. V případě nálezu prázdné hodnoty, ohlásí analytické služby jako chybu.

- *MODEL_EXISTENCE_ONLY*

  - Určuje, že daný sloupec může nabývat dvou hodnot - Missing a Existing (v případě, že je nastaven na NULL, je brán, jako by byl nastaven na Missing). Tato značka se používá u sloupců, kde je významější samotný fakt, zda je daný záznam vyplněn (Existing), nebo nevyplněn (Missing), než to, jakou hodnotu obsahuje.

- *REGRESSOR*

  - Tento parametr určuje, že daný sloupec obsahuje potenciální nezávislé proměnné (regresory). Tento parametr nezajišťuje, že bude záznam sloupce použit jako regresor (má pro algoritmus pouze dopouručující funkci).

Kromě parametrů celého DM modelu nabízí SQL Server 2008 pro každou metodu mnoho parametrů, pomocí kterých je možné optimalizovat

[1] Při psaní celé této části jsem rámcově čerpal z oficiální technické dokumentace firmy Microsoft [7]. V textu jsou pak uvedeny odkazy i přímé citace z ostatní použité literatury.

3.3 Možnosti nastavení parametrů algoritmů metod pro DM

Algoritmy metod provádějících dolování dat je v SQL Serveru 2008 možno optimalizovat pro konkrétní případ pomocí množství parametrů. Tyto parametry se liší podle typu metody. Těmito parametry lze při jejich vhodném nastavení značně ovlivnit (resp. optimalizovat) chování algoritmu.

3.4 Metody vizualizace výsledků analýzy

Každá z DM metod v SQL Serveru 2008 poskytuje nástroj pro přehledné grafické znázornění výsledků, které umožňuje sledovat např. korelace mezi vstupy a výstupy, větvení rozhodovacích stromů, příslušnost vstupních prvků k vytvořeným shlukům, atd.

3.5 Asociační metody

Asociační metody jsou používány například pro tvorbu doporučovacích systémů (kdy zákazníkovi doporučujeme určitý produkt na základě informací o jeho dřívějších objednávkách), analýzu nákupních košíků a analýze vztahů (pravidel). Tomuto využití asociačních algoritmů se počátkem 90. let věnoval Rakesh Agrawal.

V publikaci Dobývání znalostí z databází [4] nalezneme následující zápis asociačního pravidla:

$$Ant => Suc,$$

kde levá strana pravidla je předpokladem (antecedentem) a pravá strana pravidla závěrem (sukcedentem).

Agrawal [12] podrobněji definuje formální asociační pravidlo jako implikaci

$$X => I_j,$$

kde X je podmnožina prvků z množiny aributů I. I_j je samostatný prvek z množiny I, který není obsažen v X.

Pro charakterizaci asociačních pravidel používá dvě veličiny: *podpora* (support) a *spolehlivost* (confidence) - viz [4].

Podpora je hodnota vyjadřující počet objektů splňujících předpoklad i závěr :

$$P(Ant \wedge Suc) = \frac{a}{a+b+c+d}.$$

Spolehlivost je hodnota vyjadřující podmíněnou pravděpodobnost závěru, pokud je předpoklad platný:

$$P(Suc \mid Ant) = \frac{a}{a+b}.$$

V téže publikaci je uvedeno dělení asociačních pravidel podle platnosti a pokrytí (Holseheimer, Siebs, 1994):

- *Konzistentní pravidla* - platnost je rovna 1, levá strana je postačující podmínkou pro splnění pravé strany.

- *Úplná pravidla* - pokrytí je rovno 1, levá strana je nutnou podmínkou pro splnění pravé strany.
- *Deterministická pravidla* - platnost i pokrytí je rovno 1, levá strana je nutnou a postačující podmínkou pro splnění pravé strany.

Nalezneme zde rovněž zmínky a odkazy na další charakteristiky asociačních pravidel.

3.5.1 Implementace asociačních metod v SQL Serveru 2008

Asociační algoritmus prochází množinou dat a vyhledává prvky, které se vyskytují ve stejném záznamu. Asociované prvky jsou následně seskupeny do množin prvků, jejichž minimální velikost je definována parametrem *MINIMUM_SUPPORT*.

Na základě těchto množin jsou dále generována pravidla, která se později použijí pro predikci přítomnosti prvku, odvozené od přítomnosti jiných prvků, které algoritmus shledá důležité.

Samotný asociační algoritmus je implementován jako přímá implemenace Apriori algoritmu. Mimo tohoto algoritmu je pro hledání asociací možné použít také algoritmus rozhodovacích stromů - jejich výsledky se však mohou lišit. Zatímco u rozhodovacích stromů jsou pravidla tvořena na základě získaných nových informací, v případě asociačních modelů jsou pravidla založena pouze na hodnotě spolehlivosti. Pravidlo, které má vysokou spolehlivost totiž nemusí nezbytně nutně vést k vytvoření nové informace.

Algoritmus generuje kandidátní množiny, jejichž prvky představují události, produkty, a pod. Nejčastěji jde o binární hodnoty, jako ano/ne, chybějící/existující, atd. Pro každou z množin je pak vygenerováno hodnocení její podpory a spolehlivosti. Atributy obsahující spojité hodnoty jsou diskretizovány nebo seskupeny do "košů"*(buckets)*.

Číslo vyjadřující počet záznamů, obsahující požadované hodnoty (resp. jejich kombinaci) se označuje jako důležitost (někdy také frekvence). Do modelu jsou tedy zařazovány pouze ty množiny, které mají tuto hodnotu dostatečně vysokou.

Taková množina prvků, která obsahuje větší množství kombinací prvků, než je stanoven práh definovaný parametrem MINIMUM_SUPPORT se nazývá *frekventovaná množina prvků* (frequent itemset). Pokud sestává množina z prvků {A, B, C} a hodnota parametru MINIMUM_SUPPORT je např. 10, musí být každá samostatná hodnota A, B, C nalezena nejméně v 10 záznamech, aby byla začleněna do modelu - totéž platí také pro kombinaci prvků {A, B, C}.

Množství prvků obsažených v záznamech je možno vyjádřit také procentuálně, v takovém případě nastavíme parametr MINIMUM_SUPPORT na hodnotu mezi 0 (odpovídá 0%) a 1 (odpovídá 100%). Do modelu pak budou zařazeny ty záznamy, které obsahují alespoň dané procento požadovaných prvků, resp. množin.

Práh pro přípustnost pravidla je vyjádřena jako pravděpodobnost. Pokud se například množina prvků {A, B, C} objevuje u 50 záznamů, ale stejně tak se u jiných 50 záznamů objevuje množina {A, B, D} a v jiných 50 záznamech množina {A, B}, nemůžeme označit množinu {A, B} za zřejmý prediktor prvku C. Počet pravidle vytvořených modelem můžeme omezit parametrem MINIMUM_PROBABILITY.

Každému vytvořenému pravidlu je přiřazena hodnota vyjadřující jeho *důležitost* (importance). Ta se počítá jinak pro pravidla a jinak pro množiny prvků.

Důležitost množin prvků se počítá jako pravděpodobnost množiny prvků a celkového počtu záznamů. Mějme množinu obsahující prvky {A, B}. Analytické služby nejprve spočítají všechny záznamy obshaující tuto kombinaci A a B a tu pak vydělí celkovým počtem záznamů a normalizuje pravděpodobnost.

Důležitost pravidel se počítá jako pravděpodobnost pravé strany pravidla na základě jeho levé strany. Například u pravidla $If(A)Then(B)$ je nejprve spočítán poměr mezi záznamy obsahující A a B a záznamy obsahující B bez A. Poměr je pak normalizován pomocí logaritmické škály.

Tento algoritmus neprovádí žádnou formu automatické selekce rysů. Místo ní je potřeba využít nastavení parametrů. Tímto můžeme vyřadit příliš běžné prvky a události (snížením hodnoty MAXIMUM_SUPPORT), nebo naopak události a prvky s příliš nízkou mírou výskytu (zvýšením hodnoty MINIMUM_SUPPORT), případně filtrovat nepodstatná pravidla (zvýšením hodnoty MINIMUM_PROBABILITY).

3.5.2 Parametry pro optimalizaci asociační metody

Vykonávání algoritmu, zejména pak tvorba množin prvků a počítání korelací může být velice zdlouhavé. Jeho výkon je možné pozitivně ovlivnit správným nastavením řady parametrů.

Největšími překážkami v rychlém provádění algoritmu může být příliš rozsáhlá množina dat obsahující velké množství samostatných prvků, nebo například příliš nízká hodnota parametru minimální velikosti množiny prvků.

MAXIMUM_ITEMSET_COUNT

- Specifikuje maximální počet množin prvků, který má být vyprodukován. Pokud není číslo zadáno, je použita výchozí hodnota.

Výchozí hodnota je 200000.

MAXIMUM_ITEMSET_SIZE

- Specifikuje maximální množství prvků, které mohou být v jedné množině. Nastavení na 0 určuje, že velikost množiny je nelimitována.

Výchozí hodnota je 3.

MAXIMUM_SUPPORT

- Specifikuje maximální počet záznamů, které mohou být použity pro množinu prvků. Tento parametr slouží k odstranění prvků, které se objevují často a tím pádem mají malý potenciální význam.

Pokud je tato hodnota nastavena na 1, reprezentuje hodnota procento ze všech záznamů. Hodnoty větší než 1 reprezentují absolutní počet záznamů, které může množina prvků obsahovat.

Výchozí hodnota je 1.

MINIMUM_IMPORTANCE

- Specifikuje rozsah důležitosti pro asociativní pravidla. Pravidla s důležitostí nižší než tato hodnota jsou odfiltrována. (K dispozici pouze v edici Enterpise)

MINIMUM_ITEMSET_SIZE - Specifikuje minimální počet záznamů, které mohou být použity pro množinu prvků. Zvýšení tohoto čísla může v modelu snížit počet množin prvků. Lze tak například ignorovat jednoprvkové množiny.

Výchozí hodnota je 1.

MINIMUM_PROBABILITY - Specifikuje minimální pravděpodobnost, že je pravidlo pravdivé. Například nastavení této hodnoty na 0,5 znamená, že nebude generováno žádné pravdilo s pravděpodobností nižší než 50%.

Výchozí hodnota je 0,4.

MINIMUM_SUPPORT

- Specifikuje minimální počet záznamů, které musí množina prvků obsahovat, než algoritmus vygeneruje pravidlo. Nastavením této hodnoty na méně než 1 je jako minimální číslo vypočítáno jako dané procento z celkového počtu záznamů.

Nastavením na celé číslo vyšší než 1 docílíme toho, že se bude toto brát jako absolutní počet záznamů, které musí množina prvků obsahovat. Algoritmus může v případě nedostatku paměti tuto hodnotu automaticky zvýšit.

Výchozí hodnota je 0,03. To znamená, že aby byla množina prvků začleněna do modelu je nutné, aby byla nalezena nejméně ve 3% záznamů.

OPTIMIZED_PREDICTION_COUNT

- Definuje počet prvků pouitých pro optimalizaci predikce. Výchozí hodnota je 0. V takovém případě bude algoritmus produkovat takové množství predikcí, jaké muje zadáno v dotazu.

Nastavením na nenulovou hodnotu, bude predikční dotaz vracet nejvýše tolik prvků, kolik je tato hodnota, a to i v případě, že požadujete více predikcí. Každopádne může nastavení tohoto parametru zlepšit výkon predikcí.

Pokud nastavíme hodnotu například na 3, bude algoritums pro predikce používat pouze 3 prvky. Ostatní predikce, které přitom mohou být stejně pravděpodobné, neuvidíte.

3.6 Časové řady

Časové série jsou metodou, která je používána v situacích, kdy je potřeba provést odhad vývoje hodnoty určité proměnné v čase (např. vývoj prodejnosti produktu, atd.). Nalezení tohoto trendu se odvíjí od zpracování množiny základních historických dat a vytvoření modelu. Na základě těchto vstupních dat se následně provádí predikce vývoje trendu.

V publikaci *Time series analysis* [11] je pak stochastická časová série formálně definována jako nekonečná řada

$$..., X_{-2}, X_{-1}, X_0, X_1, X_2, ...$$

náhodných hodnot, nebo vektorů .

Spojení zdrojových dat a výsledků predikce se pak označuje jako řada (série). Pro správný průběh analýzy touto metodou je potřeba mít v záznamech zpracovávané databáze sloupec dat, která určují časové období, pro které záznam platí.

3.6.1 Implementace časových řad v SQL Serveru 2008

Microsoft SQL Server 2008 poskytuje pro práci s časovými řadami dva odlišné algoritmy. Prvním je, už ve verzi 2005 obsažený, algoritmus ARTxp a druhým je algoritmus ARIMA, který je nově přidaný v SQL Serveru 2008.

Ve výchozím stavu jsou pro trénování modelu používány oba algoritmy separovaně - pro získání optimálních predikcí se pak jejich výstupy míchají. Je však možné i použití pouze jednoho z algoritmů, či nastavení poměru mezi algoritmy.

3.6.1.1 ARTxp Algoritmus ARTxp (autoregressive tree algorithm) je výsledkem vývoje Microsoft Research a je založen na algoritmu rozhodovacích stromů .

ARTxp algoritmus také narozdíl od algoritmu ARIMA podporuje tzv. kříženou predikci. Pokud použijeme pro trénování algoritmu dvě oddělené příbuzné řady, je možné použít výsledný model pro predikci výsledku jedné časové řady na základě chování jiných řad (využitelné např. pro případy, kdy prodejnost jednoho produktu ovlivňuje prodej druhého). Křížená predikce je využitelná také pro tvorbu hlavního modelu, který je použit pro tvorbu dalších řad.

Algoritmus ARTxp je vhodný pro predikci následujícícho kroku (stavu).

3.6.1.2 ARIMA Modely ARIMA (autoregressive integrated moving average) jsou jednou z nejvýznamějších tříd modelů pro analýzu časových řad. ARIMA je definována lineárními relacemi mezi pozorováními a šumovými faktory. Definice procesu ARIMA v publikaci [11] zní:

Časová série X_t je procesem ARIMA (p, d, q) v případě, že $\bigtriangledown^d X_t$ je stacionární ARMA (p, q) proces.

Algoritmus ARIMA je optimalizován pro dlouhodobé predikce.

3.6.2 Parametry pro optimalizaci časových řad

Oba algoritmy podporují tzv. detekci sezónnosti, případně periodicity - pro tento účel je použita rychlá Fourierova transformace.

AUTO_DETECT_PERIODICITY

- Specifikuje detekci periodicity. Nabývá hodnot od 0 do 1. Výchozí hodnota je 0,6. Pokud je hodnota blízká 0, je periodicita detekována pouze pro silně periodická data. Nastavení hodnoty blízko 1 podporuje procházení mnoha vzorů, které jsou téměř periodické a automatické generování stop periodicity.

Zahrnutí velkého množství stop periodicity bude mít tendenci vést k výraznému zvýšení času pro trénování, ale model bude mnohem přesnější.

COMPLEXITY_PENALTY

- Kontroluje růst rozhodovacího stromu. Výchozí hodnota je 0,1.

Snížením této hodnoty se zvyšuje pravděpodobnost větvení. Zvýšení tuto pravděpodobnost naopak sníží. Tento parametr je k dispozici pouze v edici Enterprise.

FORECAST_METHOD

- Určuje, který algoritmus bude použit pro analýzu a predikci. Možné hodnoty jsou: ARTXP, ARIMA nebo MIXED.

Výchozí hodnota je MIXED.

HISTORIC_MODEL_COUNT

- Specifikuje počet historických modelů, které mají být vybudovány. Výchozí hodnota je 1. Tento parametr je k dispozici pouze v edici Enterprise.

HISTORICAL_MODEL_GAP

- Specifikuje časové prodlevy mezi dvěma po sobě jdoucími historickými modely. Výchozí hodnota je 10. Tato hodnota reprezentuje počet časových jednotek, kde jednotka je definována datovým modelem.

Například nastavením této hodntoy na "g"budou historické modely budovány pro data spadající do časových úseků v intervalech g, 2g, 3g atd.

INSTABILITY_SENSITIVITY

- Kontroluje bod, kde predikce odchylky přesáhne určitý práh a algoritmus ARTXP predikci potlačí.

Výchozí hodnota je 1.

Tento parametr se aplikuje pouze pro algoritmus ARTXP a neovlivní tedy modely používající algoritmus ARIMA. Při použití MIXED modelu je aplikován pouze pro část modelu používající algoritmus ARTXP.

Výchozí hodnota je 1. Toto nastavení poskytuje pro modely ARTXP stejné chování jako SQL Server 2005. Analytické služby pro každou predikci monitorují normalizovanou standardní odchylku. V okamžiku, kdy tato odchylka přesáhne tuto hranici, vrátí algoritmus hodnotu NULL a zastaví proces predikce.

Nastavení na 0 zastaví detekci nestability. To znamená, že můžete vytvářet i nekonečný počet predikcí bez ohledu na odchylky.

Tento parametr může být modifikován pouze v edici Enterprise. V SQL Serveru Standard je použitelná pouze výchozí hodnota 1.

MAXIMUM_SERIES_VALUE

- Specifikuje maximální hodnoty použité pro predikci. Tento parametr se používá spolu s parametrem MINIMUM_SERIES_VALUE pro omezení predikce na určité očekávané rozmezí. Například můžeme určit, že predikované množství transakcí prodeje pro kterýkoliv den by neměl přesáhnout počet produktů v inventáři.

Tento parametr je k dispozici pouze v edici Enterprise.

MINIMUM_SERIES_VALUE

- Specifikuje minimální hodnotu, která může být predikována. Tento parametr bývá používán spolu s MAXIMUM_SERIES_VALUE pro omezení predikce určité očekávané rozmezí. Například můžeme určit, že predikované množství transakcí prodeje nemůže být záporné číslo.

Tento parametr je k dispozici pouze v edici Enterprise.

MINIMUM_SUPPORT

Specifikuje minimální počet časových úseků, který je potřebný pro vytvoření větvení ve stromu každé časové řady. Výchozí hodnota je 10.

MISSING_VALUE_SUBSTITUTION

Specifikuje způsob, jakým se vyplňují mezery v historických datech. Ve výchozím stavu nejsou mezery v datech vůbec povoleny. Následující tabulka obsahuje výčet možných hodnot tohoto parametru.

Previous - Opakuje hodnotu z předešlého časového úseku. Mean - Používá pohyblivý průměr časových úseků užitých v tréninku. Numeric constant - Používá pro náhradu chybějících hodnot specifikované číslo. None (výchozí) - Nahrazuje chybějící hodnoty hodnotami z křivky trénovacího modelu.

Pokud data obsahují více řad, je vyloučeno, aby měly ”otrhané”konce. Všechny řady by měly mít stejný počáteční a konečný bod.

Analytické služby tuto hodnotu používají pro vyplnění mezer v nových datech při provádní operace PREDICTION JOIN.

PERIODICITY_HINT

Poskytuje algoritmu stopy o periodičnosti dat. Například pokud se prodeje liší každý rok a měrnou jednotkou je měsíc, je periodičnost 12. Parametr má formát {n, [,n]}, kde *n* je jakékoliv kladné číslo.

N v "[]"závorkách je nepovinné a může být opakováno podle potřeby. Například pro stopy vícečetné periodičnosti dat ukládaných každý měsíc můžeme zadat {12, 3, 1} - vzory tak budeme detekovat pro rok, kvartál a měsíc.

Periodičnost má silný vliv a kvalitu modelu. Pokud se zadané stopy liší od aktuální periodičnosti, mohou být výsledky nepříznivě ovlivněny.

Výchozí hodnota je 1.

Použití závorek je povinné. Tento parametr má datový typ *string* a pokud je zadán jako část příkazu Data Mining Extension (DMX), je nutné jej uvést v uvozovkách.

PREDICTION_SMOOTHING

- Specifikuje "promíchanost"modelu pro optimalizaci předpovědi. Můžete zde zadat jakoukoliv hodnotu mezi 0 a 1, nebo použít následující hodnoty.

Popis hodnot:

0 - určuje, že predikce používá pouze ARTXP. Předpovídání je optimalizováno pouze pro méně predikcí.

1 - určuje, že predikce bude používat pouze ARIMA. Předpovídání je optimalizováno pro více predikcí.

0,5 (výchozí) - určuje, že mají být použity oba algoritmy, a výsledky mají být smíchány.

Pro kontrolu tréninku použijte parametr FORECAST_METHOD.

Tento parametr je k dispozici pouze v edici Enterprise.

3.7 Shlukovací metody

Shlukovací metody (též analýza shluků) slouží ke zpracovávání vícerozměrných dat (objektů s více než jednou proměnnou) a nacházení podobností mezi takovými daty.

Pomocí těchto metod je možno ze vstupní množiny objektů vytvořit na základě jejich vzájemné podobnosti tzv. shluky. Metody shlukování jsou nejběžněji používaným způsobem *bezdozorového učení* (unsupervised learning) - viz [8].

V publikaci [8], jsou uvedeny následující tři hlavní cíle analýzy shluků, cituji:

- *Popis systematiky*, jenž je tradičním využitím shlukové analýzy pro průzkumové cíle a taxonomii, což je empirická klasifikace objektů.

- *Zjednodušení dat*, kdy analýza shluků poskytuje při hledání taxonomie zjednodušený pohled na objekty.

- *Identifikace vztahu*, kdy po nalezení shluků objektů, a tím i struktury mezi objekty, je snadnější odhalit vztahy mezi objekty.

Procesy shlukové analýzy můžeme rozdělit na základě požadavků na formu výsledných shluků i podle použitých metod podle několika kritérií (pro podrobnější informace o kritériích shlukování viz [1] a [2]).

3.7.1 Rozdělení podle vlastností shluků

Prvním způsobem, jakým můžeme rozdělovat metody shlukování je způsob zařazování prvků do vytvářených shluků. Shlukování, kdy je každý prvek spojen s právě jedním shlukem, nazýváme *disjunktní*. Naopak, pokud prvek (nebo jeho části) může náležet více shlukům, mluvíme o *překrývajícím se shlukování*.

Algoritmy, které vytvářejí disjunktivní shluky označujeme jako *hrubé* (hard clustering algorithms), algoritmy tvořící překrývající se shluky označujeme jako *jemné* (soft clustering algorithms).

Dalším kritériem je fakt, zda shluky vytvářejí určitou hierarchii, nebo ne. Metody, které tvoří plochou množinu shluků bez jakékoliv explicitní struktury označujeme jako *ploché shlukování* (flat) nebo také nehierarchické shlukování. Tyto algoritmy jsou nedeterministické a jako vstup vyžadují specifikaci počtu vytvářených shluků.

Výstupem hierarchických algoritmů jsou hierarchie shluků *(dendrogramy)* - takové výstupy mají větší informační hodnotu, než jsou výstupy plochého shlukování. U tohoto typu algoritmů není vyžadována specifikace počtu výstupních shluků a většina z nich se chová deterministicky. Tento fakt je bohužel vykoupen nižší efektivitou (složitost těchto algoritmů je nejméně kvadratická, zatímco ploché shlukování má lineární složitost).

3.7.2 Implementace shlukování v SQL Serveru 2008

SQL Server 2008 má pro shlukové metody dolování dat dvě metody - EM shlukování, (která se řadí mezi jemné metody) a metodu K-průměrů (spadající mezi metody hrubé).

3.7.2.1 EM shlukování EM shlukování je v SQL Serveru 2008 používána jako výchozí shlukovací metoda - oproti metodě K-průměrů poskytuje několik výhod. Jde zejména o jeho nezávislost na limitech paměti, schopnost používat pouze-dopředné kurzory [2] a překonání způsobů vzorkování. Tento algoritmus je navíc pouze "jednoprůchodový".

Algoritmus provádí iterativní tříbení dat ze vstupního shlukového modelu a prověřuje výši pravděpodobnosti, že daný datový bod náleží jednotlivým shlukům (což může vést k nepřesnostem při sumarizaci - body mohou být redundantně započítány pro každý shluk, ve kterém jsou obsaženy. Výsledky dolovacího modelu jsou tomu však přizpůsobeny). Algoritmus končí ve chvíli, kdy pravděpodobnostní model odpovídá skutečným datům.

[2] Jde o nejrychlejší updatovatelný typ kurzoru v SQL Serveru 2008. Podporuje pouze dopředný sériový přístup k záznamům. Díky toho není možné přistupovat k záznamům, které již byly kurzorem "přejety". Použití tohoto typu kurzoru je opodstatněné především v případech, kdy je nejvyšší prioritou rychlost vyhledávání a míra využití paměťového prostoru.

Pokud jsou za běhu algortimu vygenerovány prázdné shluky, případně shluky, které obsahují menší množství datových bodů než je stanovený práh, dojde k "přesazení"těchto shluků a EM algoritmus je spuštěn znovu.

Výsledkem aplikace této metody je souhrn pravděpodobností příslušnosti všech dvojic datový bod - shluk. Každý bod tedy de facto náleží do všech shluků modelu, ovšem pro různé shluky s různou pravděpodobností.

SQL Server 2008 nabízí dva typy EM shlukování: *škálovatelné* (scalable EM) a *neškálovatelné* (non-scalable EM). Při použití výchozího nastavení algoritmu škálovatelného shlukování je pro počáteční sken použito prvních 50,000 záznamů. V případě, že je sken úspěšný, jsou pro model použita tato data - pokud ne, je načteno dalších 50,000 záznamů.

Tato verze EM shlukování používá lokální buffer a je tedy schopna provádět iterace rychleji než neškálovatelná verze EM shlukování (rozdíl v rychlosti může být až trojnásobný). Ve většině případů navíc nedochází ke snížení kvality výsledného modelu.

V případě, kdy je použito neškálovatelné EM shlukování je načtena velá množina záznamů, což umožňuje zvýšit přesnost shlukovací metody, nicméně mohou výrazně vzrůst nároky na paměť.

3.7.2.2 Metoda K-průměrů (K-means)

Tato metoda je asi nejdůležitější metodou plochého shlukování. Je používána v případech, kdy je datový soubor tvořen kvantitativními proměnnými.

Metoda K-průměrů (někdy též těžišť, viz [8]) provádí začlenování objektů ze vstupního datového souboru do předem definovaného počtu shluků na základě jejich vzdálenosti od centerálních bodů (shlukových průměrů, někdy též centroidů) těchto shluků. Jedná se o tzv. algoritmus hrubého shlukování a každý datový bod je tedy začleněn do právě jednoho shluku.

Uživatel, zpravidla analytik, určí v množině dat tzv. *zárodečné body* (seed centroids), počet takto vybraných bodů - centroidů se uloží do proměnné K. Algortimus následně provede výpočty euklidovské vzdálenosti pro všechny dvojice centroid - datový uzel. Uzel je pak umístěn do shluku náležejícímu centroidu, kterému je nejblíže. Poté je pro každý existující shluk spočítán nový cenroid na základě zprůměrování hodnot bodů náležejících do daného shluku a znovu se zjišťují vzdálenosti všech bodů a následně případné přeřazování bodů do jiného shluku. Tento postup se opakuje do té doby, dokud dochází k přesunům uzlů mezi shluky.

Metoda K-průměrů poskytuje dva způsoby vzorkování množiny dat. Jde o *neškálovatelnou metodu K-průměrů* (non-scalable K-means), která najednou načte celou množinu dat a provede jeden shlukovací průchod, a *škálovatelnou metodu K-průměrů* scalable k-means, která načte prvních 50,000 záznamů a další načítá pouze v případě nutnosti.

3.7.3 Parametry pro optimalizaci shlukování

SQL Server poskytuje několik možností jak zvýšit výkon, chování a efektivitu rozhodujícího DM modelu.

CLUSTERING_METHOD

- Specifikuje, kterou shlukovací metodu má algoritmus použít. K dispozici jsou následující:

ID	Jméno metody
1	Škálovatelná EM
2	Neškálovatelná EM
3	Škálovatelná metoda K-průměrů
4	Neškálovatelná metoda K-průměrů

Výchozí hodnota je 1.

CLUSTER_COUNT

- Specifikuje přibližný počet shluků, které mají být algoritmem vygenerovány. Pokud je množství dat potřebné pro vytvoření tohoto počtu shluků nedostatečné, vygeneruje algoritmus maximální možný počet shluků. Při nastavení tohoto parametru na hodnotu 0 je pro nalezení ideálního počtu shluků použita heuristika.

Výchozí hodnota je 10.

CLUSTER_SEED

- Specifikuje jádrové číslo, které je použito pro náhodné generování shluků pro počáteční nastavení modelu.

Změnou tohoto čísla je možno měnit způsob jakým jsou budovány počáteční shluky a následně porovnat modely vybudované za použití odlišých jader.

Pokud je jádro změněno, ale nalezené shluky se příliš nezmění, je model považovatelný za relativně stabilní.

Výchozí hodnota je 0.

MINIMUM_SUPPORT

- Specifikuje minimální počet záznamů, které jsou potřeba pro vybudování shluku. Pokud je počet záznamů ve shluku nižší než tot číslo, je shluk označen jako prázdný a vyřazen.

Nastavením tohoto čísla na příliš vysokou hodnotu může způsobit vypadnutí validních shluků z modelu.

Výchozí hodnota je 1.

MODELLING_CARDINALITY

- Specifikuje počet vzorových modelů, které jsou zkonstruovány v průběhu shlukovacího procesu.

Snížením počtu kandidátních modelů může zvýšit výkon výměnou za zvýšení rizika, že bude ztracen dobrý kandidátní model.

Výchozí hodnota je 10.

STOPPING_TOLERANCE

- Specifikuje hodnotu použitou pro určení dosažení konvergence a ukončení budování modelu. Konvergence je dosaženo v okamžiku, kdy celková změna ve shlukových pravděpodobnostech je menší než průměr parametru STOPPING_TOLERANCE děleného velikostí modelu.

Výchozí hodnota je 10.

SAMPLE_SIZE

- Specifikuje počet záznamů, které algoritmus použije pro každý průchod (má smysl pouze v případě, že hodnota CLUSTERING_METHOD je nastavena na některou ze škálovatelných metod). Nastavením na 0 budou všechna data zpracována jedním průchodem. Načítání celé množiny dat může způsobit problémy s pamětí a výkonem.

Výchozí hodnota je 50000.

MAXIMUM_INPUT_ATTRIBUTES

- Určuje maximální počet vstupních atributů, které mohou být poskytnuty algoritmu před aplikováním selekce rysů. Nastavení na 0 určuje, že není žádné povolené maximum atributů.

Zvyšování počtu atributů může znatelně snížit výkon.

Výchozí hodnota je 255.

MAXIMUM_STATES

- Specifikuje maximální podporovaný počet diskrétních stavů na atribut. Pokud je počet stavů pro daný atribut vyšší než hodnota tohoto parametru, použije algoritmus pro tento atribut "nejoblíbenější"stavy a ostatní stavy ignoruje.

Zvýšení tohoto čísla může znatelně snížit výkon.

Výchozí hodnota je 100.

3.7.4 Implementace sekvenčního shlukování v SQL Serveru 2008

Tato shlukovací metoda je používána k analýze dat, která obsahují události u kterých můžeme pozorovat propojení skrze navazující "dráhy"(sekvence). Jde například o pořadí v jakém vkládá klient elektronického obchodu zboží do košíku nebo dráhu jednotlivých kliknutí uživatele při prohlížení webu. Algoritmus pak provádí shlukování vstupních objektů na základě shody těchto sekvencí.

Tento algoritmus je hybridem mezi shlukovacím EM algoritmem a analýzou Markovských řetězců [3].

[3]Markovský řetězec je pravděpodobnostní (stochastický) proces, který spolňuje tzv. Markovu vlastnost (tj. v každém stavu procesu je pravděpodobnost přechodu do dalších stavů na dřívějších stavech procesu nezávislá). Tento proces je definován dvěma parametry: vektorem absolutních pravděpodobností a maticí pravděpodobností přechodu

Specifikem této metody je použití sekvenčních dat, tedy sledů událostí v čase, případně sledů přechodů mezi různými stavy. Nejprve dojde k analýze pravděpodobnosti přechodů a následně provede měření vzájemné odlišnosti, resp. vzdálenosti všech existujících sekvencí. Na základě těchto kroků se pak vyberou sekvence vhodné jako vstupy pro shlukovací EM algoritmus.

Počet stavů, které jsou použity pro získání pravděpodobnosti současného stavu je definován řádem Markovského řetězce. Pro každý Markovský řetězec je určena přechodová matice, která obsahuje přechody pro všechny kombinace stavů. Vzhledem k tomu, že tato matice roste s přibývajícími stavy exponenciálně, dochází k rychlému nárůstu požadavků na paměť nutnou pro její uchování a na výkon CPU při zpracování.

Shlukování sleduje dva typy atributů - sekvenční a nesekvenční. Každému shluku přísluší Markovský řetězec, který zachycuje úplnou množinu "cest"a dále matice, která obsahuje sekvenci přechodů stavů a pravděpodobností. Bayesovo pravidlo pak v závislosti na počátečním rozložení určí pravděpodobnosti všech atributů daného shluku, včetně sekvencí.

MS algoritmus sekvenčního shlukování umožňuje do modelu přidávat nesekvenční atributy - ty jsou pak "přimíchány"k sekvencím, což umožní použití klasického shlukování. Model sekvenčního shlukování každopádně vede k vytvoření mnohem většího monožství shluků, než obyčejná shlukovací metoda a proto je prováděn tzv. *shlukový rozklad* (cluster decomposition), který odděluje shluky obsahující sekvence a shluky obsahující ostatní atributy.

Algoritmus sekvenčního shlukování vyžaduje několik hodnot. První hodnotou je *klíč* (single key), který jednoznačně identifikuje daný záznam.

Druhou hodnotou je samotná sekvence - jde o vhnízděnou tabulku obsahující identifikátory sloupců sekvencí (musí jít o porovnatelný datový typ - např číselný identifikátor webové stránky, textový řetězec, atd.). Pro každou sekvenci je povolen právě jeden identifikátor a pro každý model může existovat pouze jeden typ sekvencí.

Třetí, nepovinnou, hodnotou jsou nesekvenční atributy (můžou obsahovat vhnízděné sloupce).

3.7.5 Parametry pro optimalizaci sekvenčního shlukování

CLUSTER_COUNT

- Specifikuje přibližný počet shluků, které mají být algoritmem vytvořeny. Pokud nemůže být u dat toto množství shluků vytvořeno, vytvoří algoritmus tolik shluků, kolik je možné. Při nastavení tohoto parametru na 0 je pro nalezení počtu budovaných shluků použita heuristika.

Výchozí hodnota je 10.

MINIMUM_SUPPORT

- Specifikuje minimální počet záznamů potřebných pro vytvoření shluku.

Výchozí hodnota je 10.

MAXIMUM_SEQUENCE_STATES

- Specifikuje maximální počet stavů, které může sekvence mít. Nastavení tohoto čísla na hodnotu vyšší než 100 může způsobit, že algoritmus vytvoří model, který ebude poskytovat smysluplné informace.

Výchozí hodnota je 64.

MAXIMUM_STATES

- Specifikuje maximální algoritmem podporovaný počet stavů pro nesekvenční atribut. Pokud je počet stavů nesekvenčního atributu vyšší než povolené maximum, použije algoritmus "nejpopulárnější"stavy a chová se, jakoby zbývající stavy scházely (missing).

Výchozí hodnota je 100.

3.8 Rozhodovací stromy

Princip algoritmu rozhodovacích stromů je široce rozšířen i v řadě ne-informatických vědních disciplín (zejména např. v biologii). V podstatě jde o postupné rozdělování jednotlivých prvků do kateogorií, na základě jejich parametrů.

Konstrukce stromu probíhá pomocí metody *rozděl a panuj (divide and conquer)* - vstupní data se rozdělují na stále menší podmnožiny, ve kterých převládají prvky se stejnou hodnotou daného parametru. Tento postup je též označován jako *top down induction of decision trees* (indukce rozhodovacího stromu shora dolů - zkráceně TDIDT)[4].

Obecné schéma takového algoritmu je k dispozici v publikaci [4]:
algoritmus TDIDT:

1. *Zvol jeden atribut jako kořen dílčího stromu.*
2. *Rozděl data v tomto uzlu na podmnožiny podle hodnot zvoleného atributu a přidej uzel pro každou podmnožinu.*
3. *Existuje-li uzel, pro který nepatří všechna data do téže třídy, pro tento uzel opakuj postup od bodu 1, jinak skonči.*

Podstata správného průběhu algoritmu spočívá ve způsobu výběru atributu, který se použije pro větvení. Za tímto účelem jsou zpravidla používány techniky např. z oborů teorie informace a pravděpodobnosti (Shannonova entropie, informační zisk, poměrný informační zisk, vzdálenost mezi atributem a třídou [5], atd.).

3.8.1 Implementace rozhodovacích stromů v SQL Serveru 2008

Systém SQL Server 2008 podporuje tvorbu rozhodovacích stromů jak z diskrétních, tak ze spojitých atributů. Procesy tvorby a správy stromu obstarává hybridní algoritmus.

MSDT algoritmus učí Bayseovskou síť[4] pomocí předchozích znalostí a statistických dat. Část která obsluhuje metodiku výběru vhodných dat pro učení je založena na testování *podobnostní rovnosti* (likelihood equivalence).

Každý případ přebírá přednostní Bayesovskou síť a míru její důvěryhodnosti. S použitím této sítě algoritmus vypočítá pravděpodobnost a posteriori síťových struktur a vybere ty s nejvyšší hodnotou.

Metoda pro výpočet nejlepšího stromu je vybrána na základě zadané úlohy. Může jít o lineární regresi, klasifikaci nebo asociační analýzu. Tvar stromu daného modelu je závislý na ohodnocovací metodě a dalších použitých parametrech (jejich změna může ovlivnit rozdělení uzlů).

Proces tvorby stromů využívá pro určení nejhodnotnějších atributů tzv. *selekce rysů* - ta zároveň vyřazuje řídce zastoupené atributy. Jednotlivé hodnoty jsou navíc umístěny do *"košů"*, které jsou pro urychlení výkonu zpracovávány jako celek.

Strom je budován na základě rozpoznávání korelací mezi vstupy a výstupy. Po analýze všech atributů je vybrán ten, podle nějž je možno výstupy nejlépe rozdělit. Místo oddělení je určeno rovnicí počítající informační zisk - atribut s nejvyšším užitkem je použit pro rozdělení hodnot podmnožiny, na které jsou rekurentně analyzovány stejným procesem tak dlouho, dokud je strom možno dále větvit.

V případě, že jsou prediktibilní atributy i vstupy diskrétní, provádí se výpočet vytvořením matice a následným ohodnocením každé její buňky.

Pokud jsou atributy diskrétní, ale vstupy jsou spojité, je provedena automatická diskretizace všech takových vstupů.

Pro atributy diskrétní povahy, je použit klasifikační postup.

V případě atributů obsahujících spojité hodnoty je pro indikaci řezu použita lineární regrese.

3.8.2 Lineární regrese

Lineární regrese je metoda používaná pro nacházení lineárních vztahů mezi závislou a nezávislou proměnnou. Regresní křivka, která tuto relaci vyjadřuje, je definována tzv. regresní rovnicí $y = ax + b$. Kde y představuje výstupní proměnnou, x vstupní proměnnou a a a b jsou nastavitelné parametry. Tyto parametry udávají odchylku daného prvku od ideální křivky.

Z technického hlediska je algoritmus lineární regrese úpravou algoritmu rozhodovacích stromů, optimalizovanou pro modelování párových spojitých atributů. Pro tento účel jsou parametry algoritmu nastaveny tak, aby zamezily růstu stromu a data se tak držela v jediném uzlu - strom tedy netvoří žádné větve.

[4]Bayesovská síť je pravděpodobnostní model reprezentovaný acyklickým orientovaným grafem. Tento model vyjadřuje podmíněnou závislost náhodných proměnných (pozorovatelné veličiny, skryté proměnné, neznámé parametry, atd.). Hrany zde reprezentují hodnotu podmíněnou závislost - uzly, které nejsou přilehlé nevykazují žádnou závislost.

Uzel je spojen s pravděpodobnostní funkcí, jejímiž vstupy jsou příslušné vstupy pro rodičovský uzel a výstupem pravděpodobnost proměnné reprezentované vlastním uzlem [9].

Této vlastnosti je dosaženo tím, že parametr MINIMUM_LEAF_CASES je nastaven na vyšší nebo stejnou hodnotu, jako je celkový počet záznamů, které algoritmus využívá k trénování modelu. Algoritmus tak nikdy nevytvoří větvení a vykonává tedy lineární regresi.

Lineární regrese používá pro selekci rysů metodu skóre zajímavosti, je tomu tak proto, že model podporuje pouze spojité hodnoty.

Tato metoda nachází využití při předvídání vývoje trendů na základě obchodních dat, predikce výtěžku chemických reakcí, kalibraci měřících systémů, atd.

3.8.3 Parametry pro optimalizaci rozhodovacích stromů

Výkonnost algoritmu metody rozhodovacích stromů je možné ovlivnit upřesněním těchto parametrů:

COMPLEXITY_PENALTY

- Kontroluje růst rozhodovacího stromu. Nízká hodnota zvyšuje počet štěpů, vysoká jej naopak snižuje. Výchozí hodnota je založena na počtu atributů konkrétního modelu:

  - *Pro 1 až 9 atributů je výchozí hodnota 0,5.*
  - *Pro 10 až 99 atributů je výchozí hodnota 0,9.*
  - *Pro 100 a více atributů výchozí hodnota 0,99.*

FORCE_REGRESSOR

- Přikazuje algoritmu použít učené sloupce jako regresory, bez ohledu na důležitost sloupců určenou algoritmem. Tento parametr je používán pouze pro stromy predikující atributy se spojitou hodnotou.

MAXIMUM_INPUT_ATTRIBUTES

- Určuje maximální počet vstupních atributů které mohou být poskytnuty algoritmu před aplikováním selekce rysů. Nastavení na 0 selekci rysů pro vstupní atributy vyřadí.

Výchozí hodnota je 255.

MAXIMUM_OUTPUT_ATTRIBUTES

- Určuje maximální počet výstupních atributů které mohou být poskytnuty algoritmu před aplikováním selekce rysů. Nastavení na 0 selekci rysů pro výstupní atributy vyřadí.

Výchozí hodnota je 255.

MINIMUM_SUPPORT

- Určuje minimální počet listových hodnot potřebných pro vytvoření štěpu v rozhodovacím stromu.

Tuto hodnotu je vhodné zvýšit, pokud je množina zpracovávaných dat příliš velká - vyhneme se tak *přetrénování* (overtraining).

Výchozí hodnota je 10.

SCORE_METHOD

ID	Jméno
1	Entropy
2	Bayesian with K2 Prior
3	Bayesian Dirichlet Equivalent (BDE) Prior(výchozí)

SPLIT_METHOD

Určuje metodu použitou pro rozštěpení uzlu. K dispozici jsou následující možnosti.
ID Název
1 Binární: Indikuje, že strom má být rozdělen do dvou větví, nezávisle na číselné hodnotě atributu.
2 Kompletní: Indikuje, že strom může vytvářet tolik štěpů, kolik má atribut hodnot.
3 Obojí: Specifikuje, že služby analýzy (Analysis Services) mohou rozpoznat, kterou z rozdělovacích metod je vhodné použít pro získání nejlepších výsledků.
Výchozí hodnota je 3

3.9 Naivní Bayesova metoda

Bayesův teorém je pojem z teorie pravděpodobnosti (pojmenován je po Thomasu Bayesovi, který jej poprvé rozebral ve svém pojednání *An Essay towards solving a Problem in the Doctrine of Chances*). Toto tvrzení ukazuje, jak jedna podmíněná pravděpodobnost závisí na její inverzi.

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Zastoupení jednotlivých hypotéz, je vyjádřeno tzv. *apriorní pravděpodobností* P(A). Změnu pravděpodobnosti hypotézy v případě nastání události B pak vyjadřuje podmíněná pravděpodobnost $P(B \mid A)$ (též *aposteriorní pravděpodobnost*). Pravděpodobnost evidence je vyjádřena veličinou P(B).

Základním předpokladem naivního bayesovského klasifikátoru je předpoklad, že při platnosti hypotézy B jsou evidence A podmíněně nezávislé.

Bayesovská klasifikace se používá např. při filtrování spamu z E-mailových schránek, klasifikaci dokumentů, nebo pravděpodobnosti bonity klientů bank, a pod.

3.9.1 Implementace naivní Bayesovy metody v SQL Serveru 2008

Naivní Bayesova metoda (slovo naivní je zde použito proto, že nebere v úvahu také pouze potenciálně existující závislosti) je klasifikační algoritmus používaný pro prediktivní modelování.

Algortimus vychází z Bayesova teorému a je určen především pro rychlé generování dolovacích modelů a zjišťování vztahů mezi vstupními a predikovanými sloupci. Následně je vhodné na základě výsledků aplikovat další, náročnější a efektivnější metody.

Algortimus počítá pravděpodobnost všech stavů pro každý vstupní sloupec. Výstupem je množina všech možných stavů predikovaného sloupce.

Pro případné grafické znázornění je možno použít nástroj *Microsoft Naive Bayes Viewer*.

3.9.2 Parametry pro optimalizaci naivní Bayesovy metody

MAXIMUM_INPUT_ATTRIBUTES

- Určuje maximální počet vstupních atributů, které mohou být poskytnuty algoritmu před aplikováním selekce rysů. Nastavení na 0 selekci rysů pro vstupní atributy vyřadí.

Výchozí hodnota je 255.

MAXIMUM_OUTPUT_ATTRIBUTES

- Určuje maximální počet výstupních atributů, které mohou být poskytnuty algoritmu před aplikováním selekce rysů. Nastavení na 0 selekci rysů pro výstupní atributy vyřadí.

Výchozí hodnota je 255.

MINIMUM_DEPENDENCY_PROBABILITY

- Specifikuje minimální pravděpodobnost závislosti mezi vstupními a výstupními atributy. Tato hodnota je používána k nastavování limitů velikosti obsahu generovaným tímto algoritmem. Tato vlastnost nabývá hodnot od 0 do 1. Vyšší hodnoty snižují počet atributů v modelu.

Výchozí hodnota je 0,5.

MAXIMUM_STATES

- Specifikuje maximální podporovaný počet diskrétních stavů na atribut. Pokud je počet stavů pro daný atribut vyšší než hodnota tohoto parametru, použije algoritmus pro tento atribut "nejoblíbenější"stavy a ostatní stavy ignoruje.

Výchozí hodnota je 100.

3.10 Neuronové sítě

Neuronové sítě jsou zhruba 60 let starým výpočetním (respektive matematickým) modelem. Předlohou tohoto konceptu jsou biologické neuronové sítě. První matematický návrh vytvořil v roce 1943 matematik Walter Pitts ve spolupráci s neurofyziologem Warrenem McCullochem.

Na základě tohoto modelu vytvořili první elektronický obvod, který simuloval jednoduchou neuronovou síť. V roce 1949 vydal Donald Hebb svou práci "The Organization

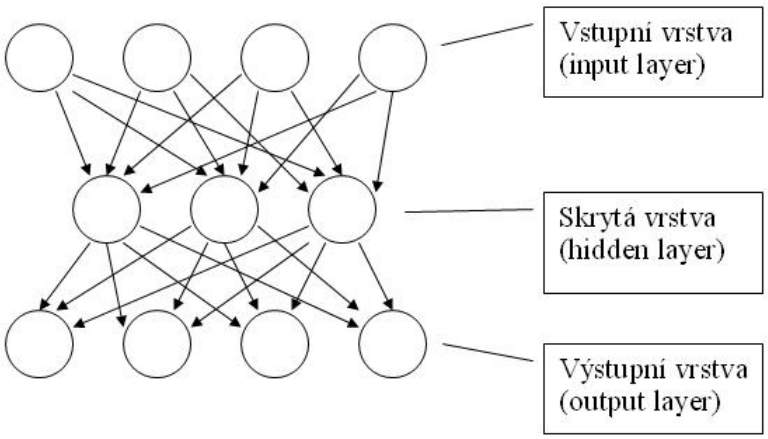
of Behavior", ve které poukazuje, že spoje mezi neurony se posilují tím více, čím více jsou používány.

Chování umělých neuronových sítí se ve své podstatě neliší od chování jejich biologických protějšků. Neuron se skládá z několika vstupních kanálů a jednoho kanálu výstupního, pomocí kterých je propojen s jinými neurony. Vstupní kanály mají nastavené váhy, pomocí kterých je neuronem vyhodnocována priorita jejich signálu.

Od doby vzniku prvních neuronových sítí, došlo k vytvoření různých specifických druhů (topologií), které se více, či méně liší ve způsobu učení, propojení neuronů, atd.

3.10.1 Implementace neuronových sítí v SQL Serveru 2008

Neuronové sítě jsou v SQL Serveru 2008 reprezentovány dvěma metodami. První metodou je model sítě vícevrstvého perceptronu, druhá metoda, logistická regrese, je principielně shodná s první - liší se však v některých technických detailech.



Obrázek 1: Schéma vícevrstvého perceptronu

3.10.1.1 Vícevrstvý perceptron MS SQL Server 2008 používá model mnohovrstvého perceptronu. Vícevrstvá síť sestává ze vstupní vrstvy neuronů, skryté vrstvy a výstupní vrstvy. Neurony stejných vrstev spolu nejsou spojeny, zato jsou propojeny se všemi neurony vrstvy následující.

Vstupní data vchází jako vstup do vstupní vrstvy a odtud se pak lavinovitě šíří do výstupní vrstvy. Vstupní neurony zprostředkovávají hodnoty vstupních atributů modelu pro dolování dat, jejich vstupem jsou tedy původní data.

Neurony skryté vrstvy zajišťují spojení mezi vstupní a výstupní vrstvou. Vstupy každého z neuronů této vrstvy jsou propojeny se všemi výstupy všech neuronů vrstvy předchozí, liší se ovšem v nastavení vah. Čím je váha vyšší, tím je tento vstup důležitější.

Váha může nabývat jak kladných tak záporných hodnot - tím je řešen problém inhibice (pro hodnoty záporné), respektive aktivace (pro hodnoty kladné) jednotlivých neuronů. Vztah mezi skrytou a výstupní vrstvou je analogický vztahu mezi vrstvou vstupní a skrytou.

3.10.1.2 Logistická regrese Algoritmus logistické regrese vychází z předchozí metody. Oproti ní je zde však hodnota parametru HIDDEN_NODE_RATIO pevně nastavena na hodnotu 0. Skrytá vrstva zde tím pádem neexistuje. Tento model tedy pracuje stejně, jako model logistické regrese.

Logistická regrese je statistickou metodou a vznikla jako alternativa metody nejmenších čtverců pro takové případy, kdy je vysvětlovaná proměnná binární. Může sloužit jako klasifikační metoda v případě, že nejsou splněny podmínky vícerozměrného normálního datového modelu.

Tato metoda odhaduje pravděpodobnost výskytu určitého jevu na základě známých hodnot, které mají na daný děj vliv (nezávislých proměnných). Pravděpodobnost, že děj nastane (tzn. sledovaná proměnná nabude hodnoty 1), je určena tzv. logistickou funkcí.

V praxi se metoda logistické regrese používá zejména v lékařství (hledání faktorů ovlivňujících výskyt infarktu, rakoviny, atd.) sociálních vědách (hledání faktorů ovlivňujících výskyt socio-patologických jevů) a průmyslu (ohodnocování výrobních jednotek).

3.10.2 Parametry pro optimalizaci neuronových sítí

HIDDEN_NODE_RATIO - Specifikace poměru mezi počtem neuronů skryté vrstvy a počtem neuronů vstupní a výstupní vrstvy. Počáteční množství neuronů ve skryté vrstvě je dáno vzorcem:

$HIDDEN\_NODE\_RATIO * SQRT(vstupní\ neurony * výstupní\ neurony)$

Výchozí hodnota tohoto parametru je 4,0.

HOLDOUT_PERCENTAGE

- Specifikace procenta záznamů z tréninkových dat použitých pro kalkulaci chyby zpoždění, která je použita jako jedno ze zastavovacích kritérií při průběhu trénování modelu.

Výchozí hodnota je 30.

HOLDOUT_SEED

- Specifikace čísla, které je použito jako jádro pseudo-náhodného generátoru pro náhodné rozpoznání pozastavovacích dat. Pokud je nastaven na 0, generuje algoritmus jádro založené na jménu dolovacího modelu.

Výchozí hodnota je 0.

MAXIMUM_INPUT_ATTRIBUTES

- Určuje maximální počet vstupních atributů které mohou být poskytnuty algoritmu před aplikováním selekce rysů. Nastavení na 0 selekci rysů pro vstupní atributy vyřadí.

Výchozí hodnota je 255.

MAXIMUM_OUTPUT_ATTRIBUTES

- Určuje maximální počet výstupních atributů které mohou být poskytnuty algoritmu před aplikováním selekce rysů. Nastavení na 0 selekci rysů pro výstupní atributy vyřadí.

Výchozí hodnota je 255.

MAXIMUM_STATES

- Specifikuje maximální podporovaný počet diskrétních stavů na atribut. Pokud je počet stavů pro daný atribut vyšší než hodnota tohoto parametru, použije algoritmus pro tento atribut ”nejoblíbenější”stavy a ostatní stavy ignoruje.

Výchozí hodnota je 100.

SAMPLE_SIZE

- Specifikuje počet záznamů použitých pro trénování modelu. Algoritmus použije buď toto číslo, nebo procentuální vyjádření získané z parametru HOLDOUT_PERCENTAGE.

Výchozí honota je 10000.

4 Experimenty s dolováním dat pomocí SQL Serveru 2008

Cílem experimentů v této páci, je demonstrace integračních a analytických služeb SQL Serveru 2008. Testování reportovacích služeb zde nemá praktický význam.

Pro experimenty byla zvolena databáze logů ze systému Moodle Slezské univerzity v Opavě. Tato databáze sestává z šesti sloupců:

Kurz - obsahuje kód určující studijní kurz

Čas - čas přihlášení do systému

IP adresa - IP adreas z které se student přihlašoval

Celý název - řetězec zastupující pro účely anonymizace skutečné jméno studenta

Akce - provedený druh akce (např. přihlášení, otevření souboru, atd.)

Informace - doplňující informace sloupce "Akce"(např. jméno otevíraného souboru, číslo prohlížené diskuze, atd.)

Soubor s databází je typu *"flat file"* (přípona .csv[5]). První řádek obsahuje názvy sloupců tabulky, dále následují jednotlivé záznamy (sloupce jsou odděleny středníky). Celkem soubor obsahuje 517 269 záznamů (49,4 MB).

Testy metod dolování dat byly prováděny na počítači obsahujícím procesor Intel Pentium Core Duo (2,8 GHz) vybaveném 2 GB RAM paměti. Pro prohlížení a manipulaci s daty zdrojového souboru se mi osvědčil program PSPad - MS Word, MS Excel a Open Office Calc nebyly schopny tento objem dat korektně zobrazit.

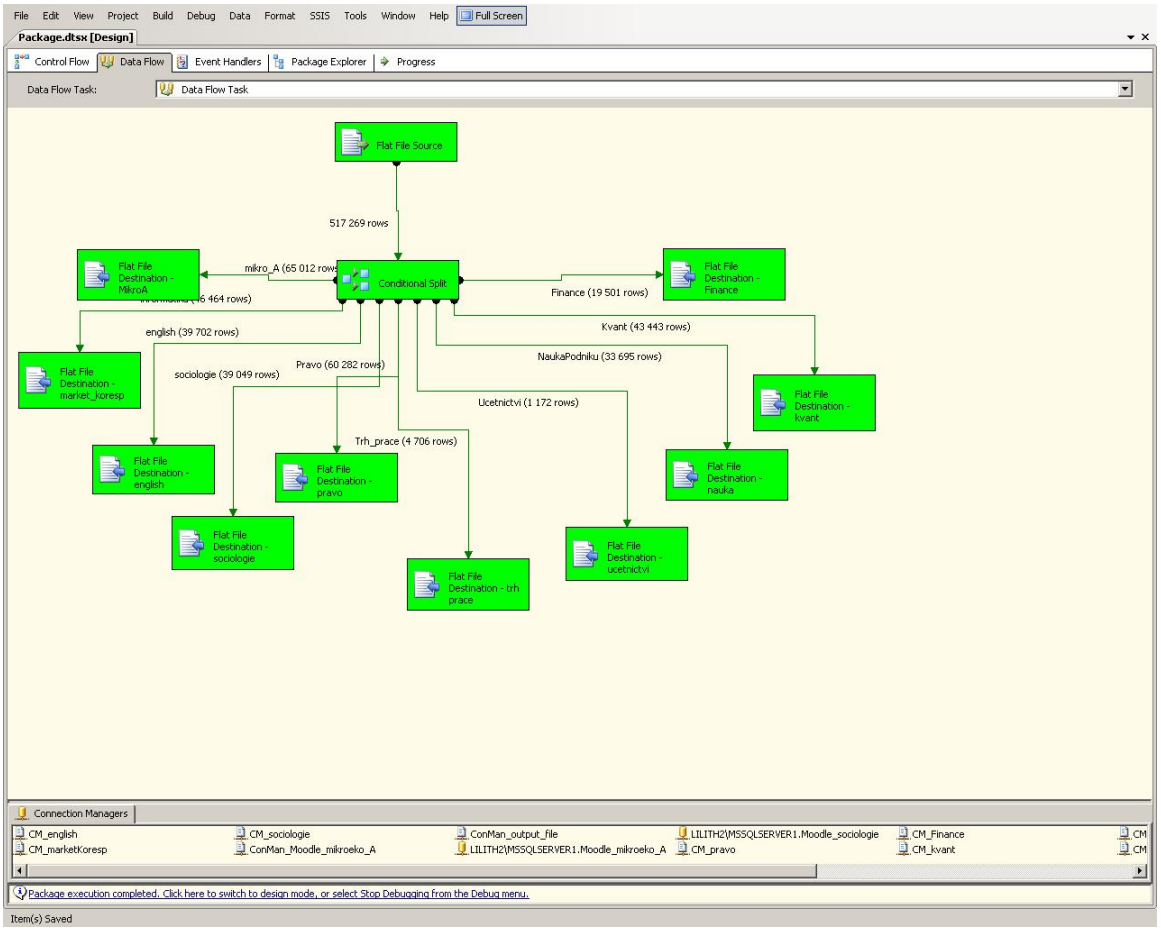
4.1 Tvorba integračního projektu

Před samotnými pokusy s aplikacemi jednotlivých DM metod je nejprve potřeba zavést data ze souboru do databáze.

Po prvním pokusu o tvorbu DM modelu nad celou databází jsem se bohužel setkal s neúspěchem - při zpracovávání databáze analytickou službou docházelo k přetečení paměti. Rozhodl jsem se proto pro extrakci záznamů do zvláštních souborů. Rozdělení jsem provedl na základě hodnoty sloupce *"Kurz"* - kurzů, kterým odpovídal relativně vysoký počet záznamů.

Vybrány byly tyto kurzy (za názvem kurzu je v závorce uveden jeho kód): Mikroekonomie A (OPF-ZS-08/09-EK/EMIA-E), Sociologie (OPF-ZS-08/09-SV/ESOC-E) a English 1 (OPF-ZS-08/09-KCJK/EA11-E).

[5]Comma Separated Values - databázový soubor s hodnotami oddělenými zpravidla čárkou



Obrázek 2: Schéma hlavního integračního projektu

4.1.1 Extrakce dat do souborů

Proces tvorby základního integračního projektu, který provádí rozdělování záznamů ze zdrojového souboru, sestával z následujících kroků:

1. V programu SQL Server Business Intelligence Developmnet Studio (v němž probíhá veškerá následující činnost) jsem vytvořil nový integrační projekt (Integration services project).

2. Dále je potřeba vytvořit cílové CSV soubory, do kterých se budou extrahované záznamy ukládat[6].

[6]Při pozdější tvorbě manažerů spojení s těmito soubory jsem zjistil, že je vhodné do těchto souborů ručně vložit názvy sloupců získané ze zdrojového souboru (tedy první řádek). Manažery spojení měly v některých případech s kopírováním názvů sloupců ze zdrojového souboru problém.

3. V novém projektu jsem na panel *Control Flow* umístil prvek *Data Flow Task*. Tato úloha se dále skládá z bloku pro načtení dat, jejich roztřídění a následné uložení do nových *CSV* souborů.

4. Další část tvorby pokračovala na panelu *Data Flow*, který slouží k sestavení jednotlivých kroků úlohy toku dat. Nejprve je potřeba určit zdroj extrahovaných dat, pro tento účel jsem použil box *"Flat File Source"*.

5. Pro spojení projektu se soubory a databázemi se používají tzv. *manažery spojení* (connection manager). Tento projekt pracuje pouze se soubory - je tedy potřeba vytvořit managery spojení pro zdrojový soubor a všechny výstupní soubory.

6. Následuje stěžejní část - tvorba funkční části, která provede rozdělení načítaných záznamů do příslušných výstupních souborů. K tomuto účelu slouží box *Conditional Split*. Datový vstup tohoto boxu je tvořen připojením zdrojového souboru. Ve vlastnostech této komponenty je pak možno nastavit počet výstupních "větví"a jejich podmínky. Pro každý kurz, který chceme ze zdrojového souboru extrahovat, vytvoříme novou větev s podmínkou formulovanou následovně:

   $$SUBSTRING(Kurz, 2, 22) == "OPF - ZS - 08/09 - EK/EMIA - E"$$

   Program tedy zjišťuje, zda daný zázam ve sloupci "Kurz"nabývá hodnoty "OPF-ZS-08/09-EK/EMIA-E"(v tomto případě jde o podmínku pro extrakci záznamů týkajících se kurzu mikroekonomie - podmínky pro ostatní kurzy jsou tvořeny obdobně).

7. Nakonec je potřeba vytvořit výstupní části integračního projektu. Výstup dat zaměříme do CSV souborů vytvořených ve druhém kroku. Pro tento případ jsem použil komponenty *Flat File Destination*, které se starají o zápis vstupních dat do souboru určeného manažerem spojení. Těchto komponent (a k nim příslušejících manažerů) je potřeba vytvořit tolik, kolik je výstupních souborů.

   Dále již jen zbývá napojit výstupní větve komponenty *Conditional Split* na vstupy komponent *Flat File Destination*(viz obr. 2) a spustit projekt.

4.1.2 Zavedení obsahu souboru do databáze

Po vytvoření oddělených souborů obsahujících záznamy týkajících se jednotlivých kurzů je potřeba načíst jejich obsah do relační databáze. Postup pro jednotlivé soubory se nijak neliší.

1. Pomocí nástroje SQL Server Management Studio jsem založil novou databázi, která bude sloužit pro uložení zdrojových dat do jednotlivých tabulek.

2. V SQL Server Business Intelligence Developmnet Studiu jsem podobně, jako v minulém případězaložil nový integrační projekt, sestávající z jediné úlohy - *Data Flow Task*.

3. Na panel *Data Flow* jsem umístil komponenty *Flat File Source*, která určuje zdrojový soubor s daty a *OLE DB Destination*, určující cílovou databázi.

4. Nakonec jsem vytvořil manažery spojení pro *Flat File* soubor a *OLE databázi*. Manažera spojení se souborem jsem připojil k souboru se zdrojovými daty a manažera spojení s databází jsem připojil k databázi vytvořené v prvním kroku a nechal jej vytvořit tabulku pro zaváděná data.

   Spuštěním projektu dojde k zavedení dat ze souboru do cílové databáze, resp. tabulky.

4.2 Testování analytických služeb

Po zavedení dat ze zdrojových souborů do databáze je možno přistoupit k testům jednotlivých analytických metod. Cílem následujících experimentů je porovnat efektivitu a vhodnost výstupů pro zpracování zadaného typu dat.

Pro účel testování analytických služeb jsem zvolil soubor s daty kurzu Mikroekonomie A. Tento soubor sestává ze stejných sloupců jako soubor se zdrojovými daty. Soubor obsahuje 65013 záznamů.

Tvorba projektu analytické služby je pro všechny analyzované soubory identická. Po založení nového analytického projektu je potřeba určit umístění *zdrojových dat* (Data Sources), *resp. zdrojový pohled* (Data Source Views) - k tomu účelu slouží příslušné položky panelu *Solution Explorer*. Sloupec "Celý název"jsem použil jako klíč. Sloupec "Kurz"je, díky předchozímu rozdělení záznamů do zvláštních souborů podle hodnot sloupce "Kurz", zbytečný (všechny záznamy tohoto sloupce jsou v rámci jednoho souboru stejné).

Všechny metody byly testovány s parametry nastavenými na výchozí hodnoty. Použité sloupce a veškeré ostatní odlišnosti jsou vždy zmíněny.

4.2.1 Test shlukovacích metod

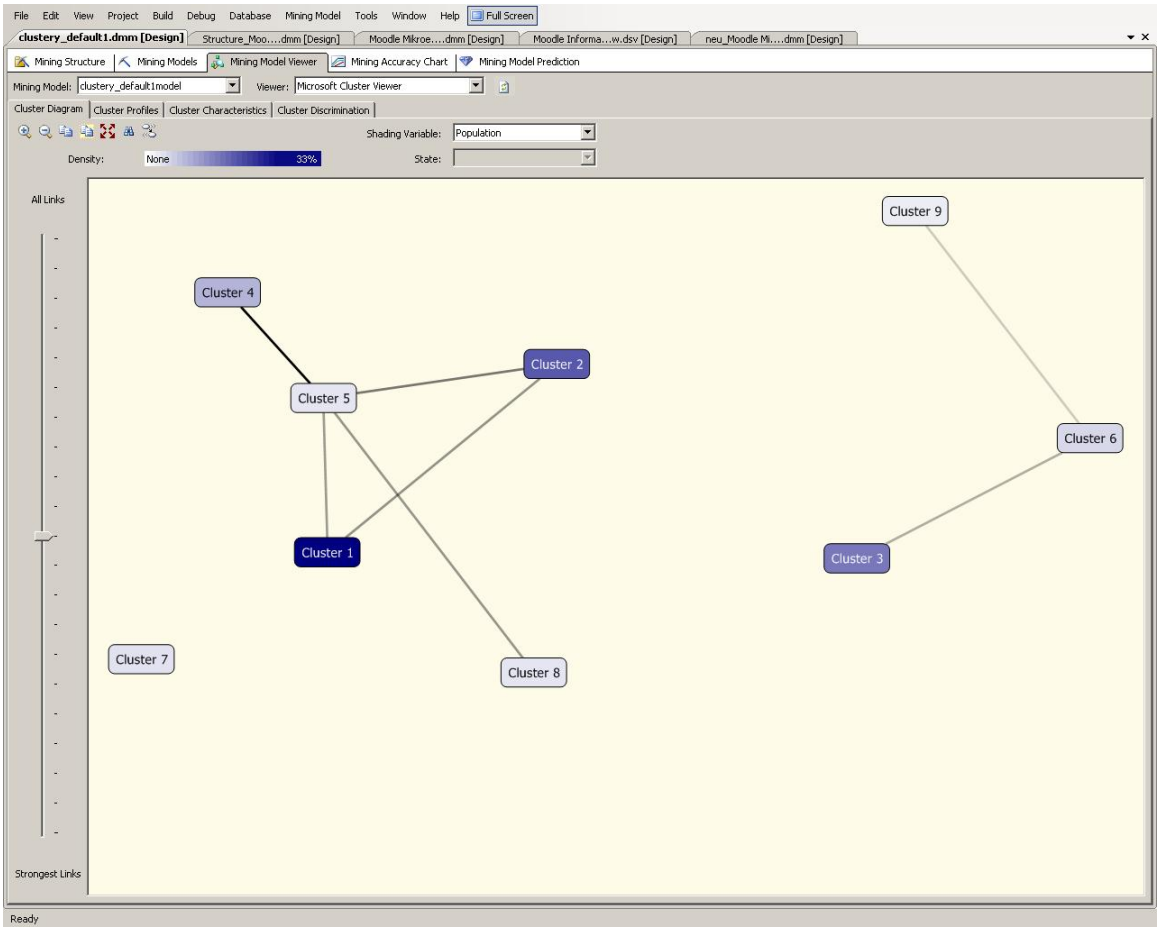
Shlukovací metodu jsem použil pro analýzu četnosti hodnot jednotlivých sloupců. Jako vstupní hodnoty jsem použil sloupce IP adresa, Čas (hodnota sloupce Čas byla diskretizována), Akce a Informace. Jako klíčový, jsem použil sloupec Celý název. Predikce nebyla požadována nad žádným sloupcem.

Analýza trvala 57 sekund.

Tato metoda dává uživateli k dispozici čtyři panely s grafickými výstupy: Cluster Diagram, Cluster Profiles, Cluster Characteristics a Cluster Discrimination.

Panel diagramu shluků (viz obr. 3) nám umožňuje vidět vztahy mezi jednotlivými shluky a sílu vazeb mezi nimi (síla vazeb je přímo úměrná intenzitě barvy spojovací hrany). Barva uzlů znázorňuje procento z celkové populace, které shluk zastupuje (rozlišování barvou uzlů se dá změnit na kterýkoliv ze vstupních atributů).

Panel s profily shluků (viz obr. 4) je z informačního hlediska ze všech nejpřínosnější. Vidíme zde zastoupení hodnot parametrů v jednotlivých shlucích. První sloupec označuje vstupní sloupec, druhý nejastěji se vyskytující hodnoty pro daný sloupec a shluk. Zbylé sloupce zachycují poměry mezi hodnotami ve shlucích.



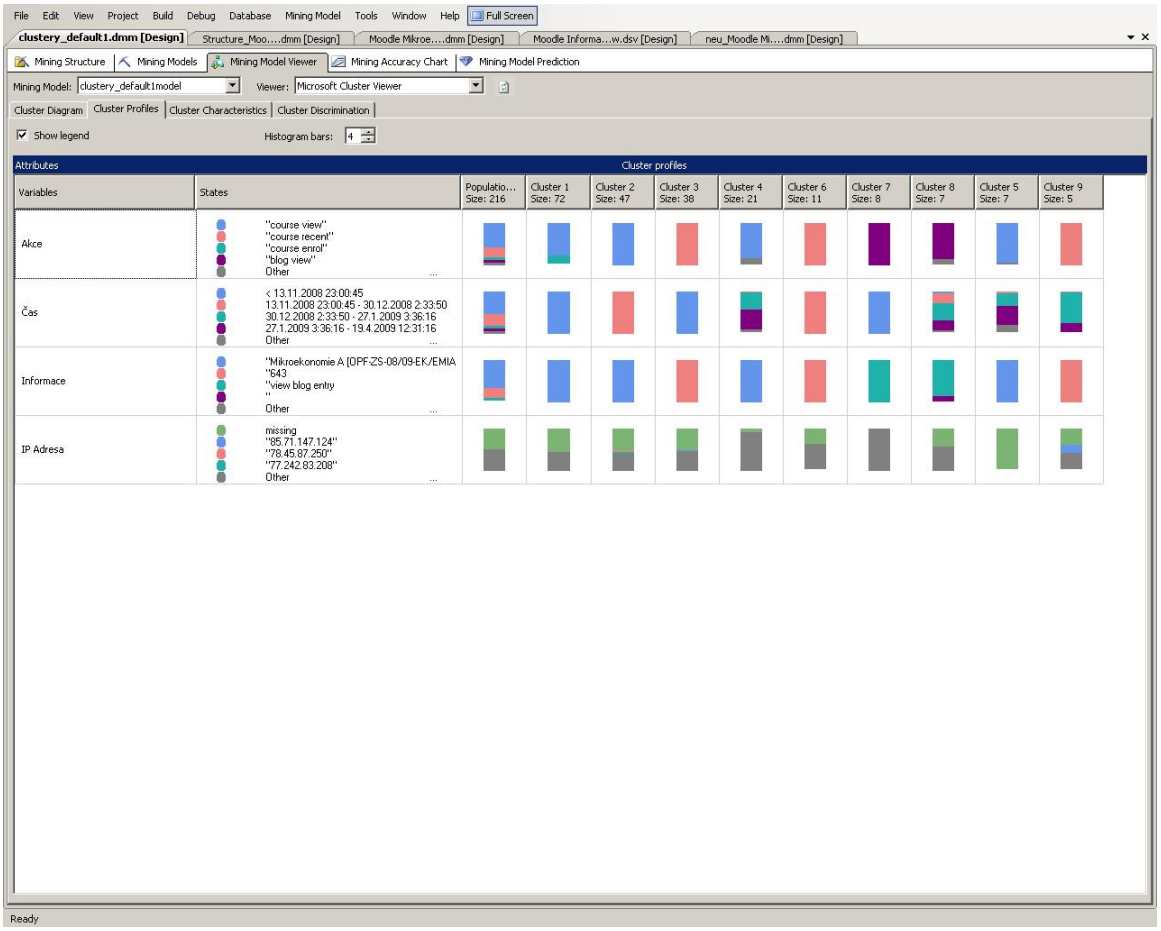
Obrázek 3: Diagram shluků (Cluster Diagram)

Vidíme, že např. nejčastěji prováděné akce jsou: prohlížení úvodní stránky kurzu, zápis do kurzu a prohlížení blogu. Drtivá většina přihlášení proběhla před 30.12.2008. Ze sloupce s IP adresami vidíme, že nejčaseji probíhalo přihlašování z adres 85.71.147.124, 78.45.87.250 a 77.242.83.208.

Panel s charakteristikou shluků (viz obr. 5) nám umožňuje procházet obsah jednotlivých shluků a zjistit, s jakou pravděpodobností nabývá záznam umístěný v daném shluku určitých hodnot.

Nástroj pro vizualizaci výsledků shlukovací metody dále obsahuje panel *cluster discrimination*, který umožňuje porovnávat dvojice zvolených shluků.

Shlukovací metody nám umožňují použít dva typy algoritmů, výchozí hodnota této metody je nastavena na algoritmus škálovatelného EM-shlukování. Rozhodl jsem se proto dále otestovat algoritmus K-průměrů (nastavením parametru CLUSTERING_METHOD na hodnotu 3). Jak je vidět z obrázku 6, došlo v rozvržení k určitým změnám.

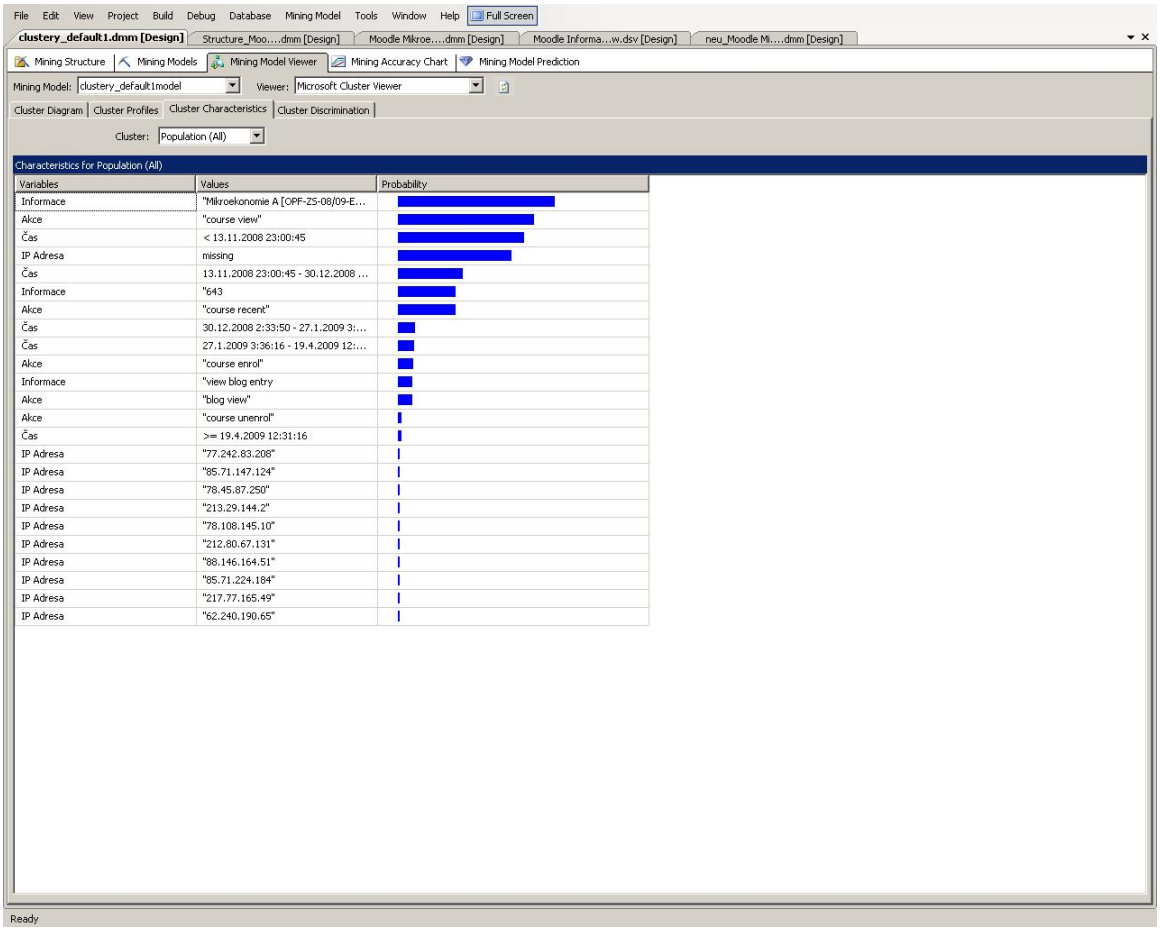


Obrázek 4: Profily shluků (Cluster Profiles)

Prakticky vymizely shluky, které by obsahovaly více než dvě hodnoty - ve sloupci *Čas* se tak shluky méně překrývají. Byly zde také nalezeny dvě nové opakující se IP adresy (217.77.165.49 a 88.146.164.51).

Dále zde byla vyselektován shluk skupiny uživatelů, kteří se začátkem semestru zapisovali do kurzu (cluster č. 3 - záznamy z tohoto clusteru byly pravděpodobně v předchozím případě zahrnuty do clusteru č. 1). Nicméně i po aplikaci odlišné metody zůstaly některé shluky zcela stejné (viz následující tabulku).

	EM-shlukování	shlukování K-mean
čísla shluků	8	9
čísla shluků	2	2
čísla shluků	6	6



Obrázek 5: Charakteristika shluků (Cluster Characteristics)

4.2.2 Test naivní Bayesovy metody

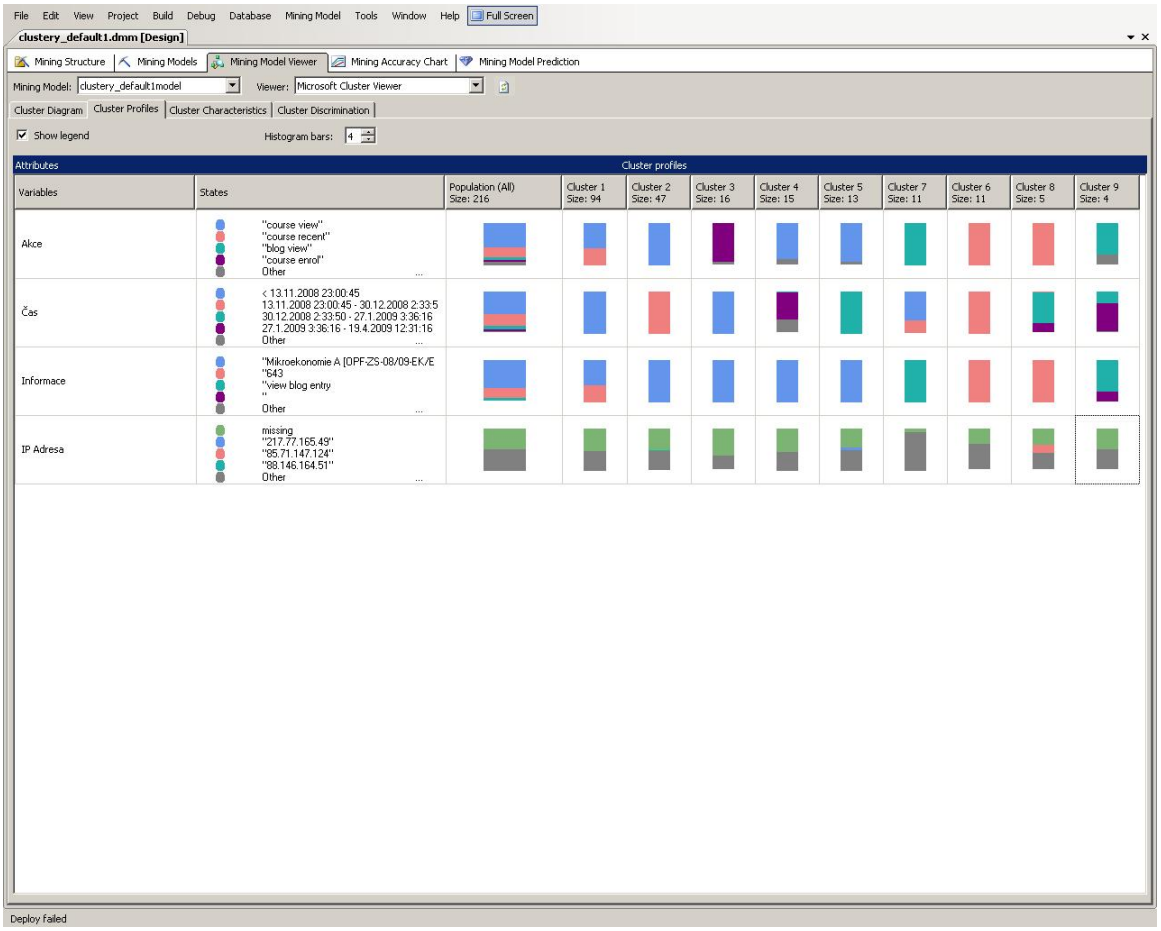
Pro aplikaci naivní Bayesovy metody jsem jako vstupní sloupce zvolil *Čas*, *Akce* a *IP adresa*, sloupec *Informace* jsem použil jako predikovaný a sloupec *Celý název* jako klíčový.

Analýza trvala 48 sekund. Vizualizace výsledků sestává ze čtyř panelů: ”Dependency network”, ”Attribute profile”, ”Attribute characteristics”, ”Attribute discrimination”.

Panel Dependency network obsahuje síť závislostí. Výsledkem této analýzy byla závislost hodnoty atributu *Informace* na hodnote atributu *Akce*. Lze tedy předpokládat, že predikovaný sloupec *Informace* není výrazně závislý na hodnotách zbylých sloupců.

Panel s profily atributů 7 zobrazuje hodnoty sloupce *Informace* a ty hodnoty sloupce *Akce*, které je vyvolávají. Není překvapivé, že v kurzu Mikroekonomie A je nejvíce zastoupena akce zobrazující úvodní stránku kurzu a k ní náleží informace (tedy parametr akce) obsahující kód kurzu Mikroekonomie A.

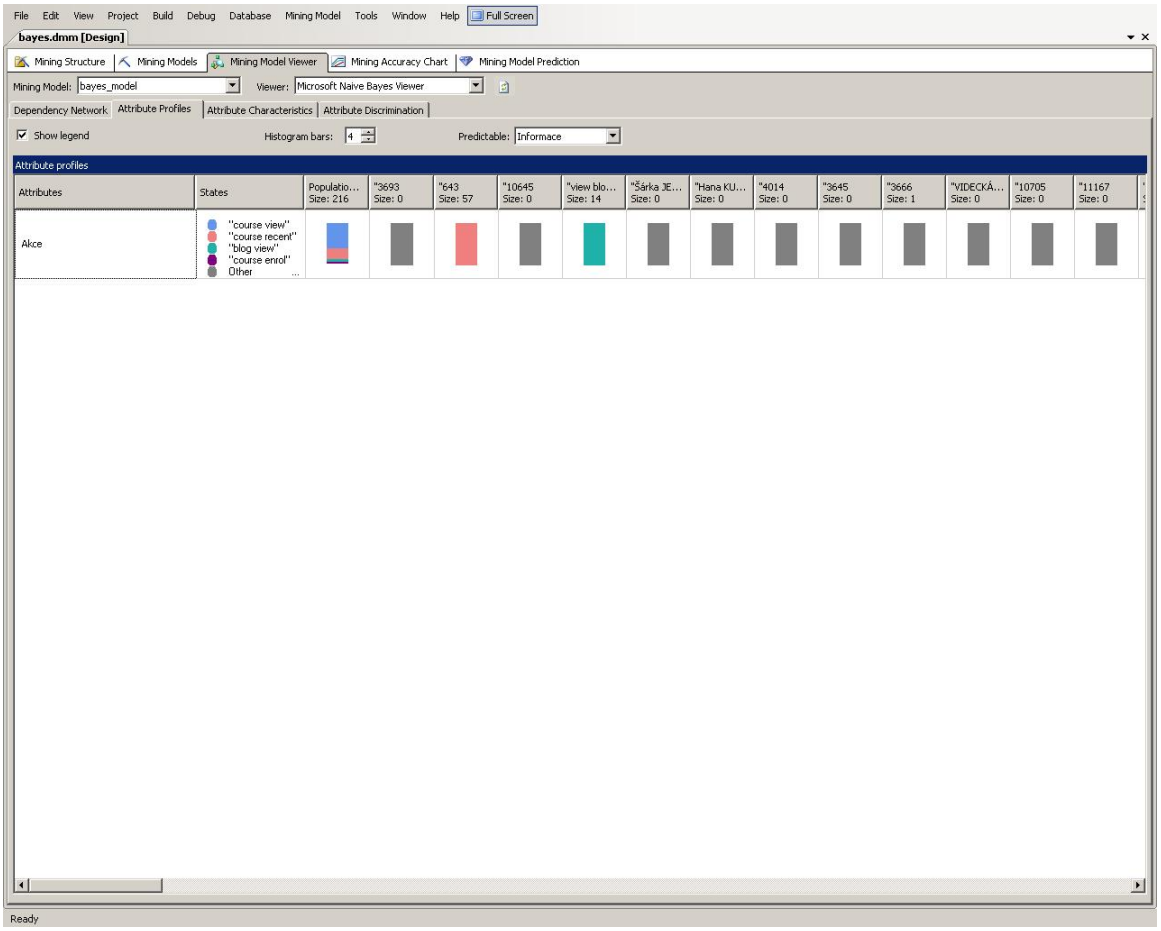
Vzhledem k tomu, že na sloupce *Akce* a *Informace* není možné aplikovat diskretizaci, je grafický výstup naivní Bayesovy metody pro toto množství hodnot poměrně nevhodný.



Obrázek 6: Charakteristika shluků (Cluster Characteristics) po použití algoritmu K-mean

Panel s charakteristikou atributů pro tuto analýzu neobsahuje žádné informace a panel diskriminace atributů nám umožňuje porovnávat u dvojic parametrů *Informace* pravděpodobnost, že se objeví v záznamu s příslušnou hodnotou *Akce*.

Mimo fakt, že hodnota sloupce *Informace* je závislá primárně na hodnotě sloupce *Akce* (a tedy hodnoty sloupců *Čas* a *IP adresa* mají celkem zanedbatelný vliv), nepřinesla analýza naivní Bayesovou metodou žádné zajímavé poznatky.



Obrázek 7: Bayesova metoda - profily atributů

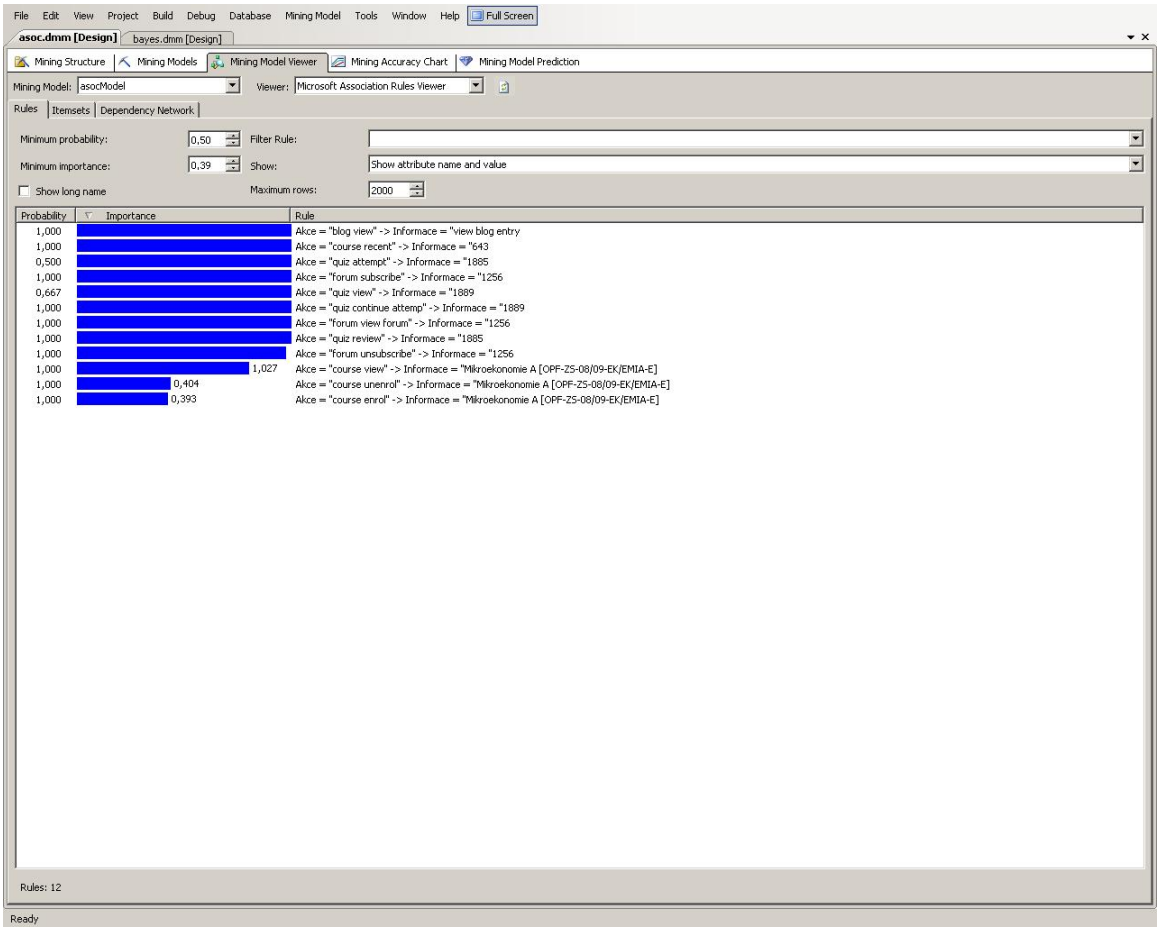
4.2.3 Test asociačních pravidel

Metodu asociačních pravidel jsem vyzkoušel dvěma způsoby. Poprvé pro zjištění pravidel závislosti mezi sloupci *Akce* (vstupní) a *Informace* (predikovaný) - jako klíč jsem opět použil sloupec *Celý název*.

Výpočet této analýzy trval 29 sekund. Výsledkem byl souhrn nalezených pravidel, ten je ve vizualizačním nástroji znázorněn v panelu *Rules* (viz obr. 8).

Tato analýza nemá, vzhledem k závislosti mezi sloupci, která byla zjištěna v předchozí analýze, význam pro hledání samotných pravidel. Nicméně můžeme díky ní vysledovat nejčastěji se vyskytující akce a k nim příslušející parametry. Pro tento účel se hodí panel *Dependency network* (viz obr. 10), který zachycuje síť závislostí *Informací* na *Akcích*. Z něj je zřejmé např. že nejvíce navštěvované je diskuzní fórum s číslem 1256. U tohoto fóra je zároveň nejvyšší výskyt požadavků o zasílání nových příspěvků na e-mail.

Dále je např. patrné, že u testu č. 1889 je nadprůměrně vysoký počet akcí ”quiz view”(označuje prohlížení podmínek testu) a ”quiz continue attempt”(označuje opako-

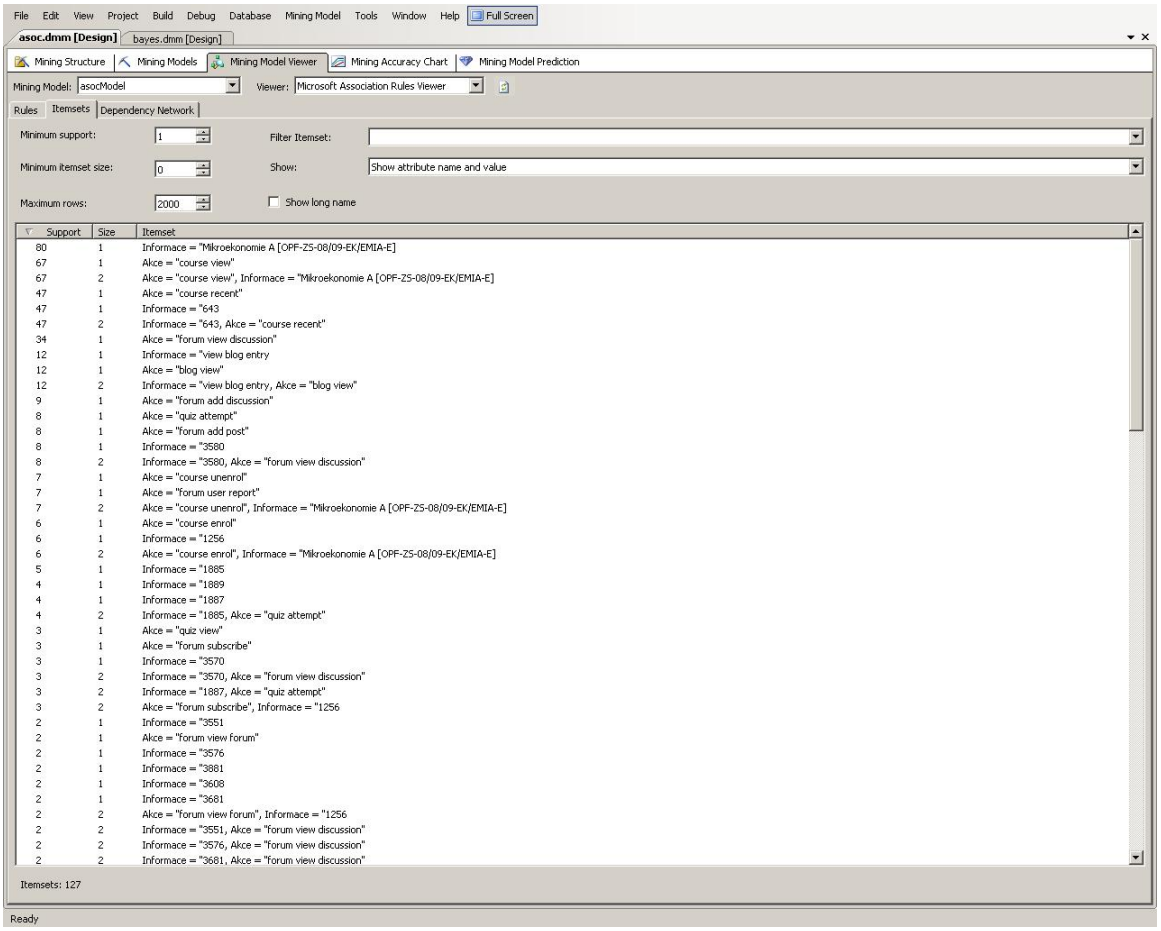


Obrázek 8: Asociační pravidla - nalezená pravidla

vaný pous o absolvování testu) - z toho můžeme vyvodit, že šlo pravděpodobně o obtížný test.

U testu č. 1885 je evidentně nadprůměrný počet akcí ”quiz review”(označuje prohlížení výsledků absolvovaného testu) a ”quiz attempt”(označuje vyhodnocení testu), což by mohlo znamenat, že tento test byl absolvován velkým počtem studentů kurzu a navíc se při jeho absolvování studentům dařilo lépe než průměrně (chybí zde vazba na akci ”quiz continue attempt”- ta se pro tento test evidentně vyskytovala jen podprůměrně).

Podruhé jsem se pokusil pomocí asociačních pravidel zjistit závislost akcí na čase. Sloupec *Čas* jse použil jako vstupní, sloupec *Akce* jako predikovaný a sloupec *Celkový název* jako klíč. Protože se dá předpokládat, že kardinalita sloupce *Čas* bude velmi vysoká (soubor obsahuje záznamy za necelý jeden rok), a výsledky analýzy by tak byly nepřehledné, rozhodl jsem se nad tímto sloupcem provést diskretizaci (akce nastavitelná při tvorbě DM procesu).

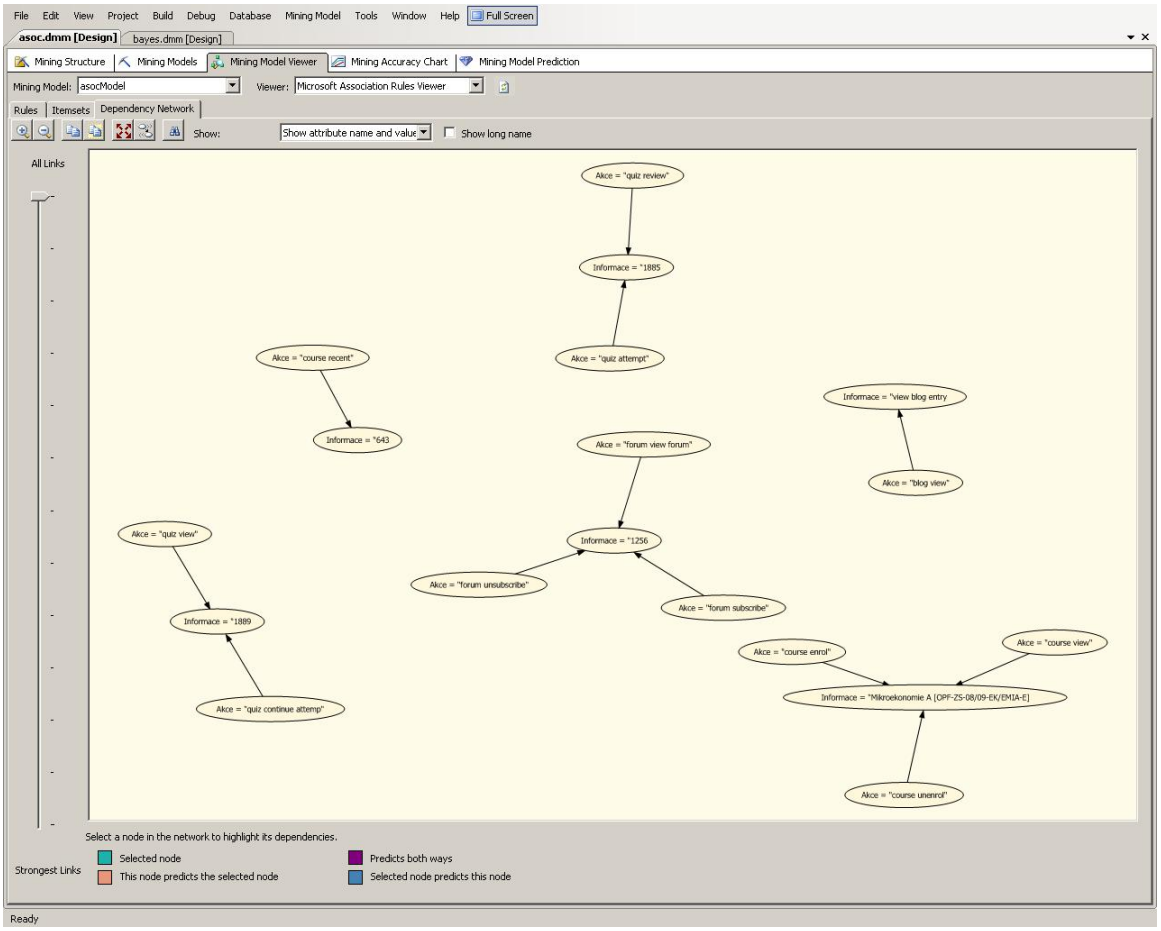


Obrázek 9: Asociační pravidla - množina nalezených prvků

Analýza trvala 31 sekund. Z vizualizace výsledků analýzy (viz obr. 11) je zřejmé, že proces diskretizace rozdělil hodnoty sloupce *Čas* na 5 období. Ze sítě závislostí je evidentní, že např. zatímco zobrazování úvodní stránky kurzu probíhá po celý rok, tak např. prohlížení blogů probíhá až od konce ledna a odhlašování z kurzu probíhá až na konci roku.

4.2.4 Test neuronových sítí

Pro test metody neuronových sítí jsem zvolil jako vstupní parametry sloupce *Akce*, *IP adresa* a *Čas* (diskretizován). Predikován byl sloupec *Akce* a jako klíč sloužil sloupec *Celý název*. Analýza proběhla za 38 sekund. MS SQL Server vybral ze sloupce *Akce* pro predikci hodnoty "quiz view"a "forum add post". Pravděpodobnost uskutečnění těchto akcí pak prováděl na základě IP adresy, případně časového období (viz obr. 12).

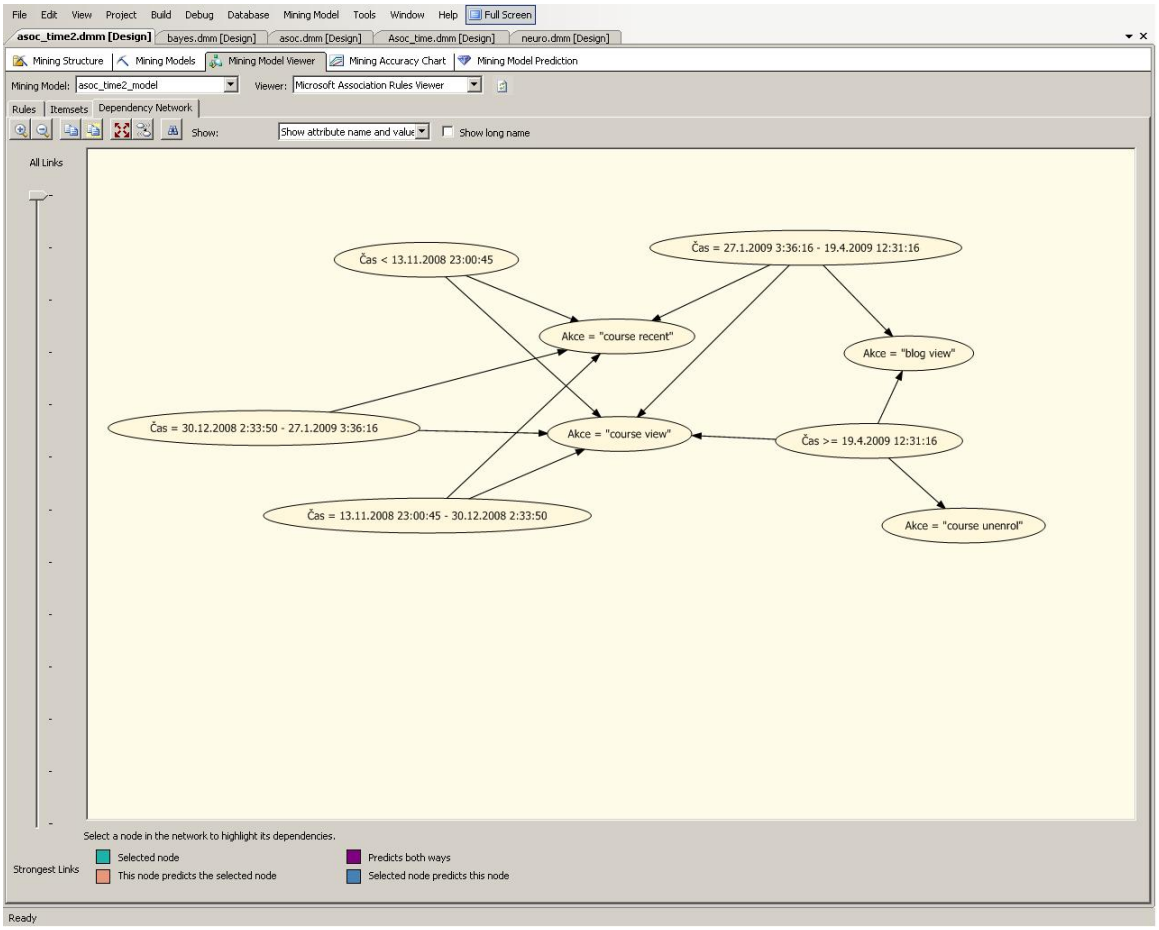


Obrázek 10: Asociační pravidla - síť závislostí informací na akcích

Dvojice hodnot, jejichž pravděpodobnost je poměřována je možno dále volit, nicméně ne vždy je možné pro všechny dvojice pravděpodobnost dopočítat a zobrazit grafický výstup.

Tato metoda pro zadaný soubor dat není příliš vhodná. Poblém vidím opět v tom, že není možné provést diskretizaci žádného sloupce kromě sloupce s časovým údajem. Výstup je navíc prezentován formou poměřování hodnot z dvojice sloupců, což v případě, že kardinalita obou těchto sloupců není malá, má ten efekt, že vizualizace výstupu této analytické metody je značně nepřehledná.

U obecné analýzy takového typu dat tedy tato metoda selhává. Vhodná by byla v případě, že bychom potřebovali analyzovat chování jednoho nebo několika málo studentů (na základě hodnoty sloupce *Celý název*), případně počítačů (na základě IP adres). Pro tento případ naopak nejsou příliš vhodné realtivně obecně zaměřené metody, jako je shlukování.



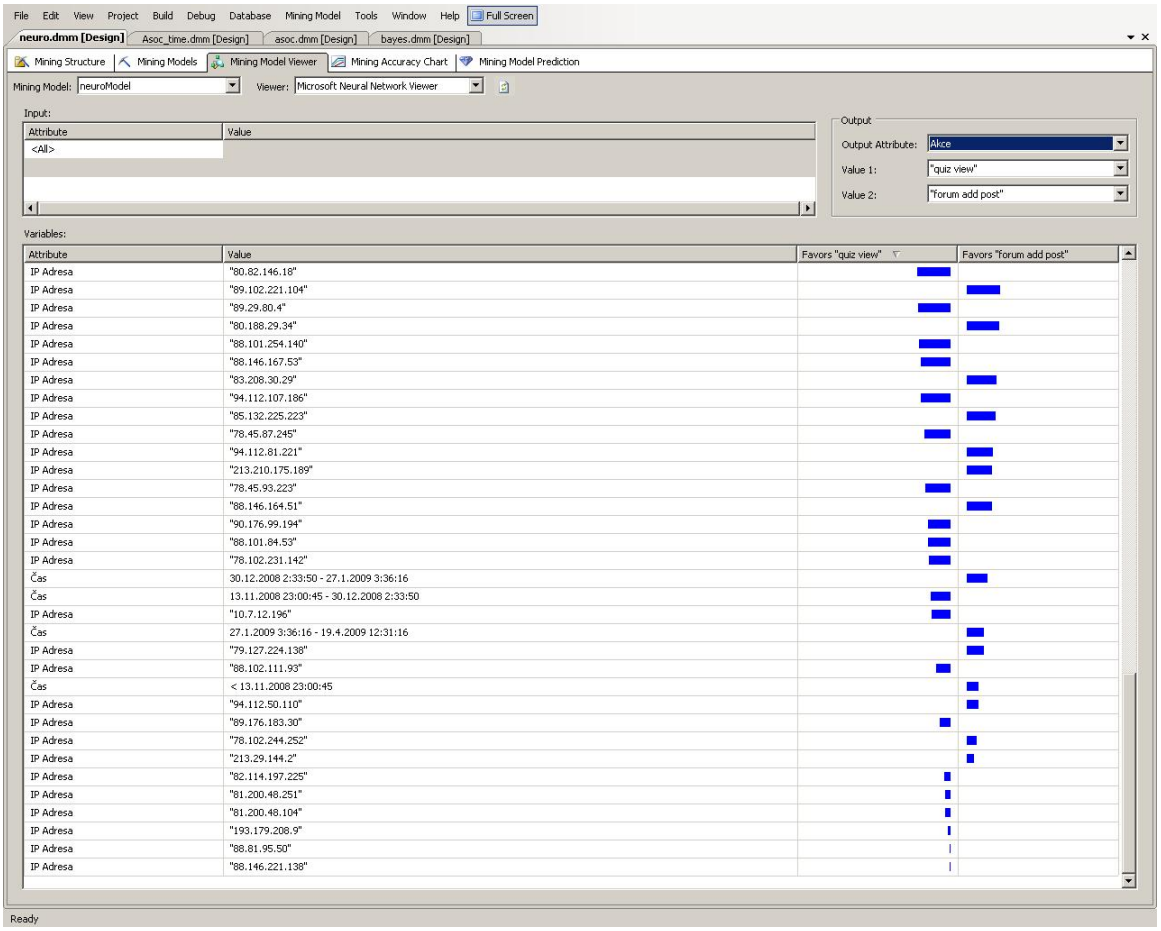
Obrázek 11: Asociační pravidla - síť závislosti akcí na čase

Dalším vhodným užitím by byla predikce sloupce s nízkou kardinalitou (v ideálním případě binární - např. sloupec obsajující logickou hodnotu podle toho, zda daný student na konci semestru úspěšně ukončil kurz, nebo nikoliv).

4.2.5 Test logistické regrese

Logistická regrese je v SQL Serveru 2008 z technického hlediska specifickou formou neuronové sítě. Testem této metody jsem se rozhodl analyzovat závislost sloupce *Akce* na *čase*. Jako klíč jsem tedy nastavil sloupec *Celý název*, jako vstupní *Čas* a jako predikovaný *Akce*.

Test se plnohodnotně podařil až napodruhé. Napoprvé selhal nástroj pro vizualizaci - důvodem byl příliš vysoký počet hodnot ve sloupci *Čas*. Podruhé jsem tedy před spuštěním analýzy nechal provést diskretizaci sloupce *Čas*. Díky tomu se viuzualizace omezila na zobrazení pěti časových období. Analýza trvala 31 sekund.



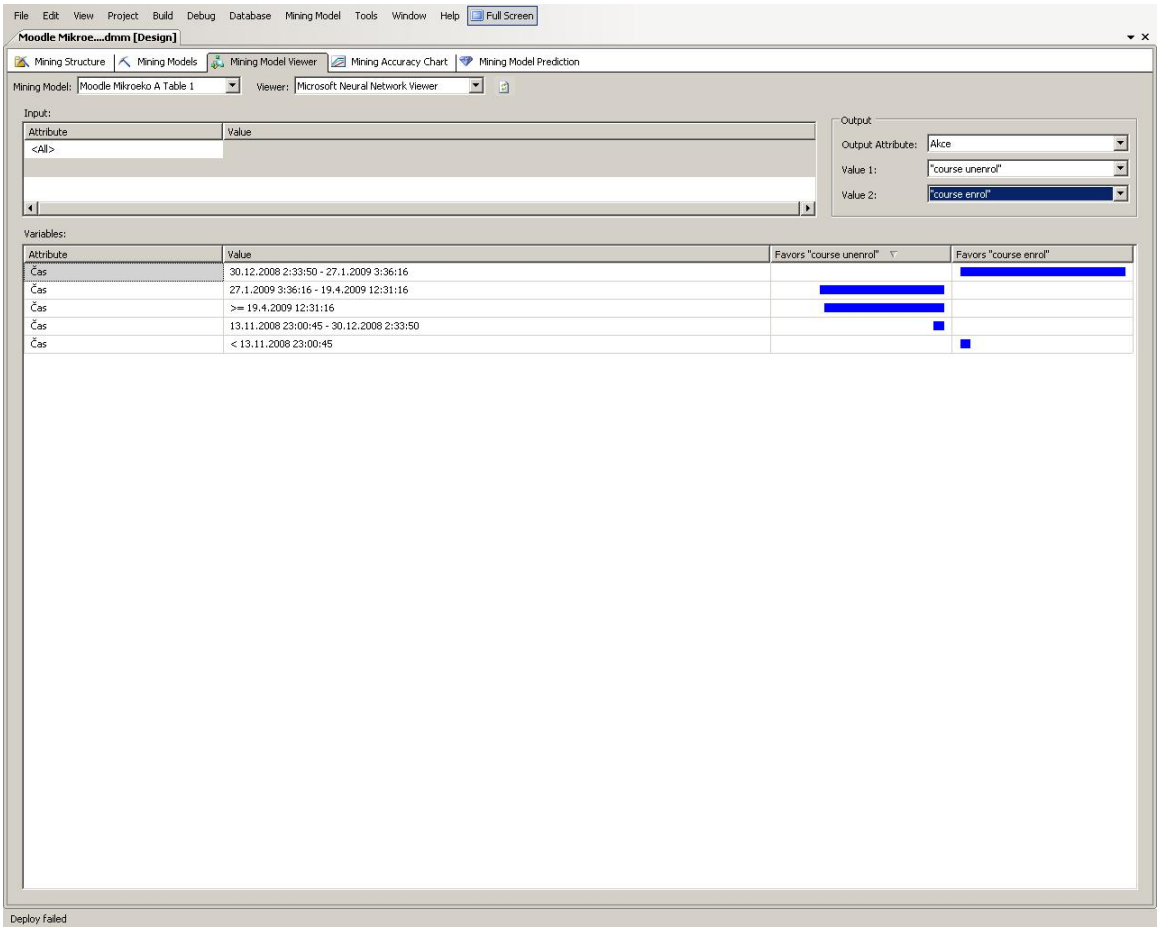
Obrázek 12: Neuronové sítě - přehled preferencí

Podobně jako metoda neuronových sítí umožňuje logistická regrese porovnávání pravděpodobnosti uskutečnění určitého jevu. V našem případě pravděpodobnost uskutečnění jedné, nebo druhé zvolené akce v průběhu jednotlivých období.

Pro demostraci jsem zvolil poměr mezi hodnotami "course enrol"a "course unenrol"- tedy poměr toho, jak se v jednotlivých obdobích lišilo přihlašování, resp. odhlašování z kurzu Mikroekonomie A. Je na první pohled patrné, že přihlašování do kurzu probíhalo nejvíce v průběhu měsíce ledna. V následujících obdobích pak probíhalo jen odhlašování (přihlašování na kurz bylo evidentně časové omezeno).

4.2.6 Test sekvenčního shlukování

Metoda sekvenčního shlukování slouží k odhalení návazností určitých prováděných akcí. Při tomto testu byly vstupními hodnotami sloupce *Čas* (opět diskretizován), *IP adresa*, *Informace* a *Akce*. Klíčovým sloupcem byl sloupec *Celý název*. Pro predikci jem nevybíral



Obrázek 13: Logistická regrese - přehled preferencí

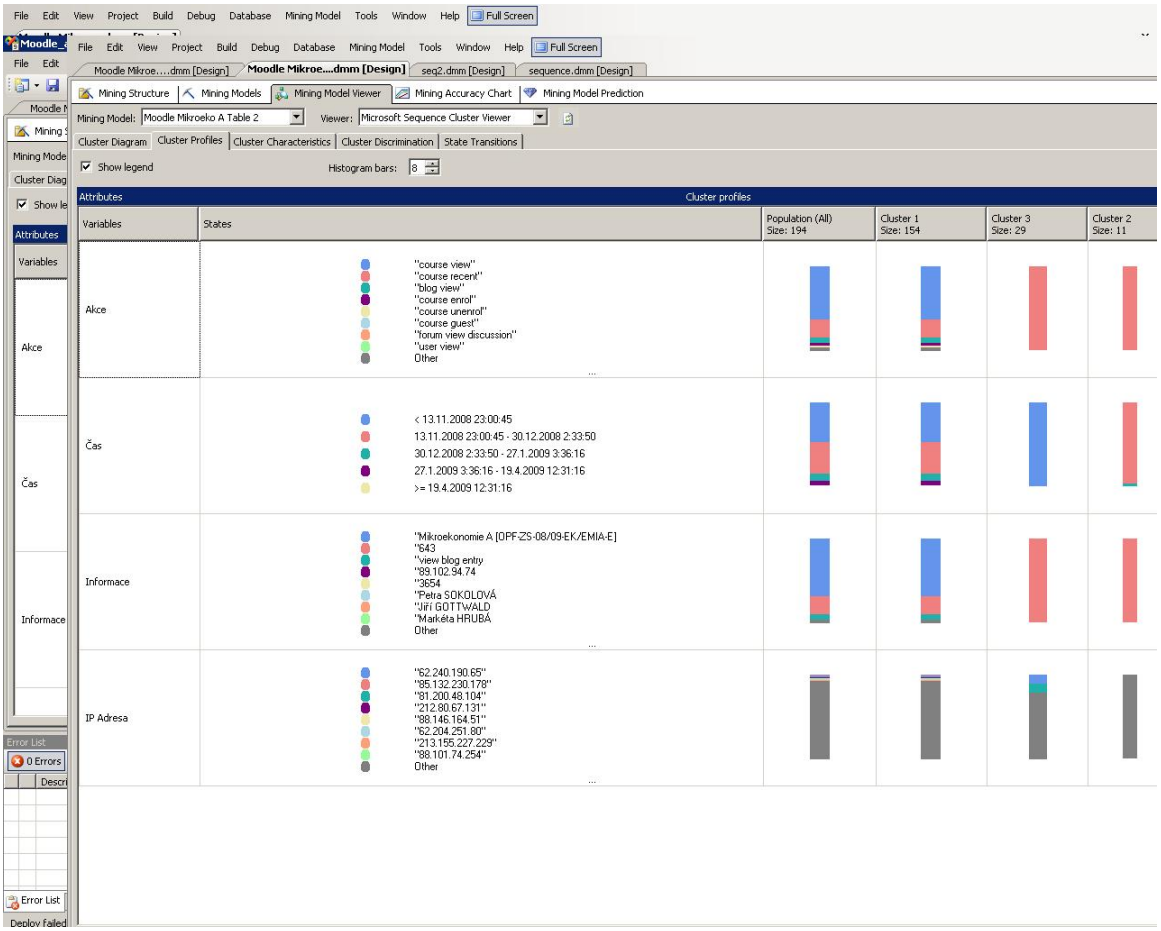
žánou hodnotu - šlo mi pouze o nalezení sekvencí, ne predikci potenciálně následujících akcí.

Délka analýzy byla 49 sekund. Algoritmus sekvenčního shlukování však ve zdrojové databázi nenalezl žádné sekvence. Můžeme tedy předpokládat, že studenti své aktivity v systému Moodle neprováděli v žádném zaznamenatelném vzoru.

Výsledkem analýzy bylo vytvoření pouhých tří shluků (viz obr. 14), v porovnání s klasickými metodami shlukování je informační přínos této analýzy jen minimální.

4.2.7 Test metody rozhodovacích stromů

Pomocí metody rozhodovacích stromů můžeme provádět dělení vstupní množiny dat na podmnožiny aplikací série kritérií. Testování této metody bylo poměrně dost problematické. S výchozím nastavením algoritmu v podstatě nebylo možné vytvořit větvení - vytvořený strom sestával z jediného uzlu.

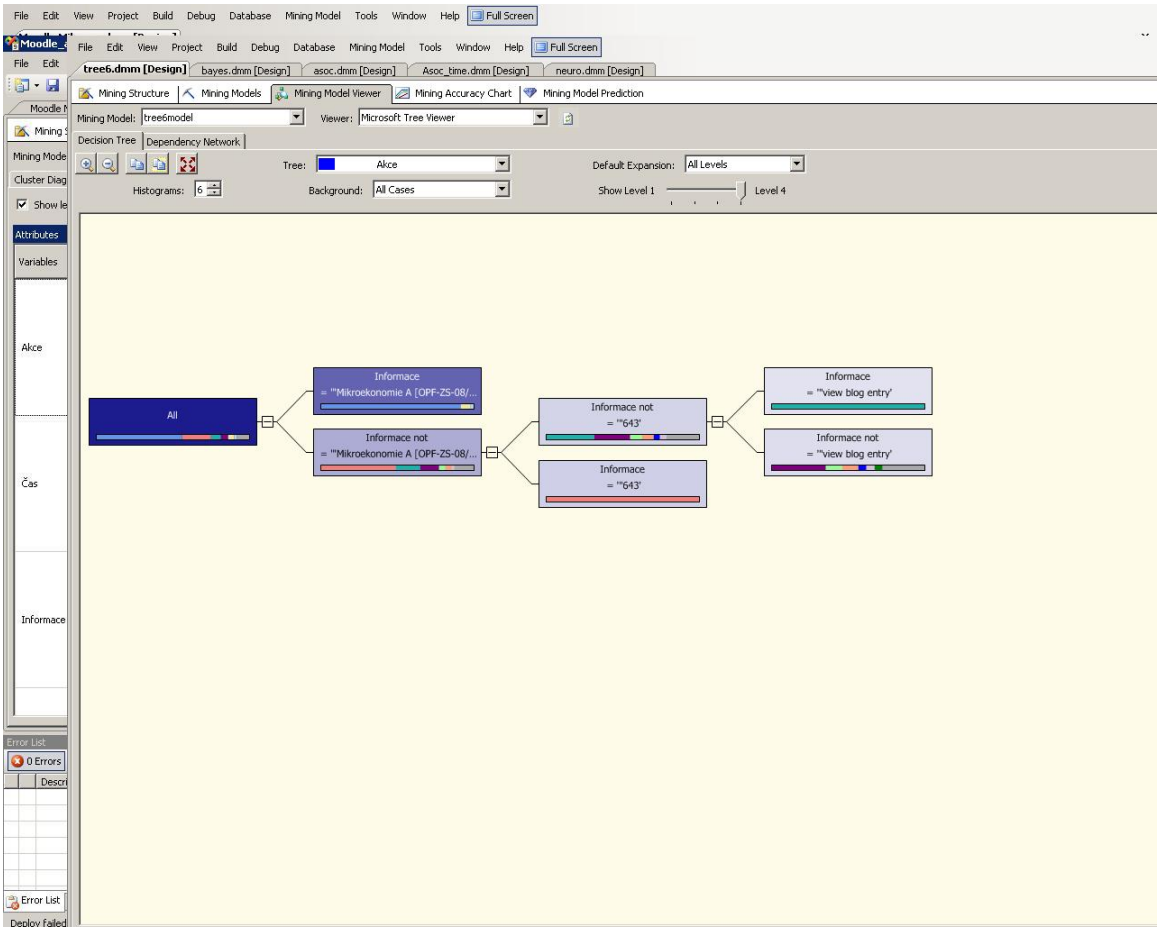


Obrázek 14: Sekvenční shlukování - profily shluků

Problém jsem vyřešil nastavením hodnot parametru *COMPLEXITY_PENALTY* (nízká hodnota zvyšuje pravděpodobnost větvení) na hodnotu 0.1 a parametru *MINIMUM_SUPPORT* (udává minimální počet položek v uzlu stromu) na hodnotu 2 (ve výchozím stavu byla nastavena na 10) Při testování této metody jsem jako vstupy použil sloupce *Informace* a *IP adresa*, jako líč sloužil sloupec *Celý název* a predikoval jsem sloupec *Akce*. Analýza probíhala 39 sekund.

Nicméně výsledky ani tak nebyly příliš uspokojivé. Algoritmus vybral ty hodnoty sloupce *Akce*, které měly významné zastoupení a na základě k nim příslušejících (resp. nepříslušejících) hodnot sloupce *Informace* prováděl binární větvení (viz obr. 15).

Nalezena byla pouhá tři hodnotící kritéria. Každý uzel pak obsahuje ty akce, které nabývají hodnoty uvedené v popisku uzlu. Takto provedená analýza tedy nepřinesla žádné užitečné informace. Je možné, že na větší množině dat by bylo možné větvení provádět snáze i při výchozím nastavení, použití významně větší vstupní databáze bohužel



Obrázek 15: Rozhodovací strom

dostupný hardware neumožňoval. Metodu rozhodovacích stromů by bylo jednodušší aplikovat na databázi obsahující převážně numerická data.

Bohužel jsem nemohl otestovat funkčnost metody časových řad a metody lineární regrese, protože kromě časového vstupu dále vyžadují další vstup, který obshauje spojité hodnoty. V testovací databázi se však žádný sloupec obsahující taková data nevyskytuje.

5 Závěr

Ve své práci jsem obecně popsal jednotlivé funkce programu MS SQL 2008, které slouží pro podporu Business Intelligence. Jedná se o integrační, analytické a reportovací služby. Ve zvláštní kapitole jsem popsal principy funkce, způsoby implementace a vhodná použití jednotlivých analytických služeb.

V části zabývající se experimenty jsem demonstroval aplikaci integračních a analytických služeb na zadaná data a vyhodnotil vhodnost jednotlivých způsobů analýzy.

Program MS SQL Server 2008 poskytuje značné množství nástrojů pro integraci datových zdrojů (např. vytvoření databáze z textového souboru, souboru tabulkového procesoru, atd.), filtraci vstupních dat, jejich dodatečnou úpravu, atd. Použití integračních služeb je poměrně intuitivní. Při práci s programem není nezbytně nutná znalost jazyka MS SQL, prakticky všechny potřebné příkazy jsou generovány automaticky.

Obsažené analytické služby jsou mocným nástrojem pro dolování dat. Pomocí nich je možné nacházet v databázích vzory, které umožnují následné predikce.

Data, která jsem obdržel pro demostraci metod dolování dat, které MS SQL Server poskytuje, jsem nejprve upravil pomocí integrčních služeb a následně je nahrál do databáze serveru.

Následně jsem na nich otestoval funkčnost jednotlivých metod. Ne všechny se ukázaly jako vhodné. Nejlepší výsledky poskytly metody shlukovací a asociační pravidla. Pomocí těchto jsem zjistil např. nejčastější hodnoty vstupních sloupců, případně vztahy mezi nimi. Pomocí metody neuronových sítí a logistické regrese pak bylo možno sledovat např. chování vybraných uživatelů, případně počítačů.

Naivní Bayesova metoda a metoda rozhodovacích stromů se ukázaly jako neefektivní, metodu časových řad a lineární regresi pak díky charakteru vstupních dat nebylo možno otestovat vůbec.

V průběhu práce s MS SQL Serverem 2008 jsem se setkal s několika nepříjemnostmi. Program občas přestal odpovídat, zejména při náhledu na panel vizualizace výsledků analýzy a při rozbalování nabídky dostupných serverů (při tvorbě manažerů spojení). Dále mi připadalo nevhodné vyhodnocování chybných nastavení projektu až při spuštění analýzy (omylem jsem nastavil diskretizaci pro sloupec obsahující text, progam při analýze zhavaroval, protože diskretizace nad tímto typem dat nebyla možná).

Domnívám se, že MS SQL Server 2008 je výborným prostředkem pro operace dolování dat, nicméně jeho metody nejsou univerzální a nemusí být pro analýzu daného typu dat stejně vhodné - je tedy potřeba je používat uvážlivě.

6 Reference

[1] ŘEZANKOVÁ, Hana; HÚSEK, Dušan; SNÁŠEL, Václav. *Shluková analýza dat* . 2. Praha : Professional Publishing, 2009. 218 s. ISBN 978-80-86946-81-8.

[2] *Introduction to Information Retrieval* [online]. 2008, 07-Apr-2009 [cit. 2010-04-26]. Dostupné z WWW: < http://nlp.stanford.edu/IR-book/>.

[3] LACKO, Ľuboslav. *Business Intelligence v SQL Serveru 2005*. 1. Brno : Computer Press, 2006. 391 s. ISBN 80-251-1110-5.

[4] BERKA, Petr. *Dobývání znalostí z databází*. 1. Praha : Academia, 2003. 366 s. ISBN 80-200-1062-9.

[5] DE MANTARAS, Lopez. *A Distance-Based Attribute Selection Measure for Decision Tree Induction*. Machine Learning [online]. 1991, 6, [cit. 2010-04-26]. Dostupný z WWW: <http://www.springerlink.com/content/hp3215h75t0054k2/fulltext.pdf>.

[6] FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. *From Data Mining to Knowledge Discovery in Databases*. AI Magazine [online]. 1997, 17, 3, [cit. 2010-03-08]. Dostupný z WWW: <http://www.aaai.org/ojs/index.php/aimagazine/article/view/1230/1131>.

[7] Microsoft. *Microsoft Developer Network* [online]. 2010 [cit. 2010-03-08]. Dostupné z WWW: <http://msdn.microsoft.com/en-us/default.aspx>.

[8] MELOUN, Milan; MILITKÝ, Jiří; HILL, Martin. *Počítačová analýza vícerozměrných dat v příkladech*. 1. Praha : Academia, 2005. 449 s. ISBN 80-200-1335-0.

[9] *Bayesian network In Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2003, 2010 [cit. 2010-03-08]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Bayesian_network>.

[10] *Markov chain In Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2002, http://en.wikipedia.org/wiki/Markov_chain [cit. 2010-03-08]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Markov_chain>.

[11] VAN DER VAART, Aad W. *Time series* [online]. Amsterdam : [s.n.], 1995-2001 [cit. 2010-05-02]. Dostupné z WWW: <http://www.researchgate.net/publication/40517175_Time_Series>.

[12] AGRAWAL, Rakesh; IMIELINSKI, Tomasz; SWAMI, Arun *Mining Association Rules between Sets of Items in Large Databases*. IBM Almaden Research Center : 650 Harry Road, San Jose, CA 95120, 1993 [cit. 2010-05-02]. Dostupné z WWW: <http://rakesh.agrawal-family.com/papers/sigmod93assoc.pdf>

[13] LACKO, Luboslav. *Databáze: datové sklady : OLAP a dolování dat s příklady v Microsoft SQL Serveru a Oracle*. 1. Brno : Computer Press, 2003. 469 s. ISBN 80-7226-969-0.

[14] Power, D.J. *A Brief History of Decision Support Systems*. DSSResources.COM, World Wide Web, http://DSSResources.COM/history/dsshistory.html, version 4.0, March 10, 2007.

[15] HURBEAN, Luminita. *Business Intelligence: applications, trends, and strategies*. Economic Sciences Series [online]. 2006, 1, [cit. 2010-05-02]. Dostupný z WWW: <http://anale.feaa.uaic.ro/anale/resurse/46_Hurbean_L_-_Business_intelligence-applications,_trends_and_strategies.pdf>.